8568A
Spectrum Analyzer
Remote Operation

AUGUST 1978

HEWLETT PACKARD

1400 Fountain Grove Parkway
Santa Rosa, California, U.S.A.

Printed in U.S.A.
08568-90003

CHAPTERS

CONTENTS

Appendices **Page**

Chapter 1

INTRODUCTION TO REMOTE OPERATION

This chapter discusses the requirements for remote operation of the spectrum analyzer using an HP-IB* computing controller.

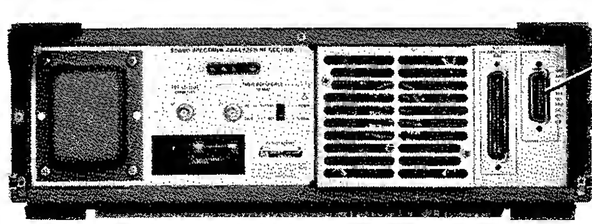
General Description

The standard HP 8568A Spectrum Analyzer is equipped for HP-IB operation. HP-IB hardware includes the HP-IB interface card and the rear panel HP-IB connector. An HP-IB interconnection cable is required to connect the analyzer to the controller HP-IB interface. Programming codes are summarized on the lower pullout information card and in a foldout inside the back cover of this manual. A more detailed syntax summary can be found in Appendix C.

HP 8568A Spectrum Analyzer

HP-IB Interconnection Cable, (supplied with the HP Controller HP-IB interface option)

INFORMATION Cards with Programing Codes



HP-IB Connector (A13J1)

RF Section Rear Panel

---

* Hewlett Packard Interface Bus, the Hewlett-Packard implementation of instrument interface standard IEEE Std. 488-1975 and ANSI Std. MC1.1, ''Digital interface for programmable instrumentation''

Remote Operation Overview

The standard 8568A Spectrum Analyzer with an HP-IB controller allows:

- **Remote operation of the analyzer's front panel functions, including the shift key functions. See Chapter 2.**
- **Output of any analyzer function value or trace amplitude. See Chapter 3.**
- **Input of special CRT display labels and graphics. See Chapter 4.**
- **Interrupt of controller for service or data transfer. See Chapter 5.**

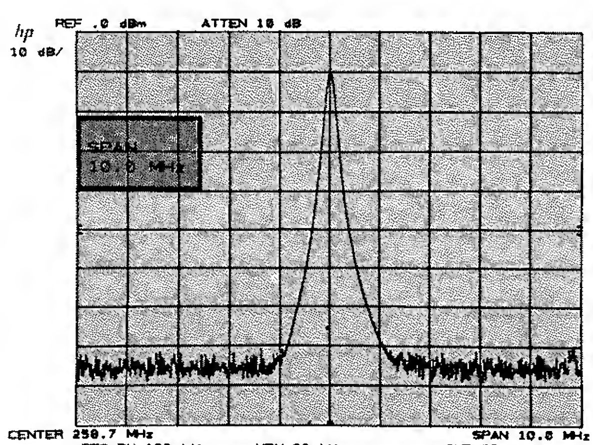
The following illustrate these programming modes:

Change Front Panel Functions

To set the center frequency to 258.7 MHz and the span to 10 MHz with the HP 9825A Computing Controller:

wrt 7 18, "IP CF 258.7 MZ SP 10 MZ"

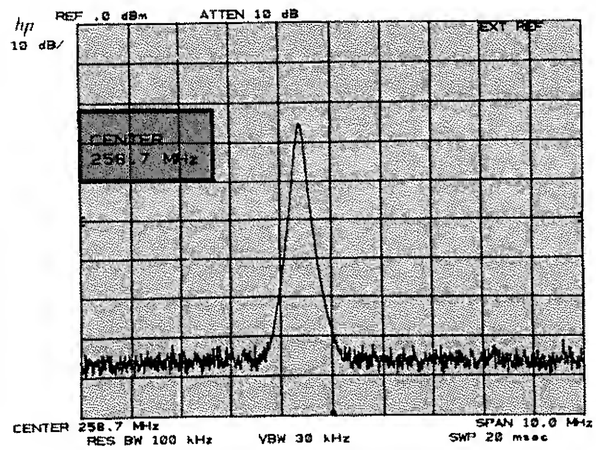
Part	Meaning
wrt	controller TALK
7	select HP-IB
18,	analyzer address
"IP	instrument preset
CF	activate center frequency
(spaces)	optional spaces inserted for clarity*
258.7 MZ	set center frequency value
SP	activate frequency span
10 MZ"	set frequency span value



* The spaces' omission will not affect the program.

Output Value or Amplitude

To output the center frequency into controller variable F and display F, first activate center frequency, then execute



wrt 718, "OA"

output active parameter

red 7 18, F; dsp "center frequency =", F

controller LISTEN
select HP-IB
analyzer address
store frequency (Hz) in F
display F on desktop controller

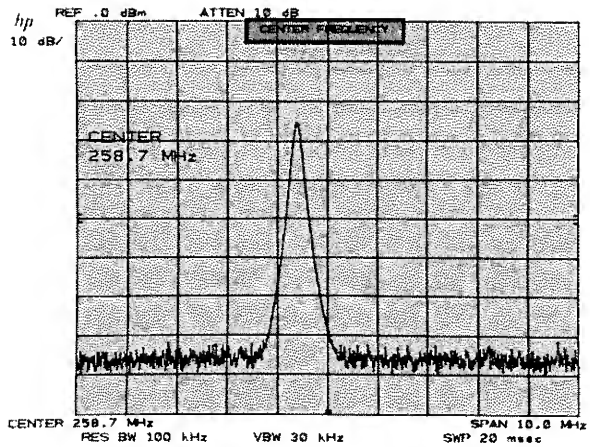
9825 Display reads "center frequency = 258700000.00".

Input CRT Labels and Graphics

To write "Center Frequency" in the top center of the graticule area:

wrt 718, "EM PU PA 448 , 928 LB CENTER FREQUENCY"; wtb 718,3

clears trace C memory
Pen up
Plot absolute vector to:
x position
required delimiter
y position
The following ASCII characters are a label
Text
Label terminator



HP-IB Controller

Any HP-IB compatible controller can be used to operate the 8568A. The overall system measurement speed and capability depends, to a large extent, on the computing, storage and interrupt capabilities of the controller.

The HP 9825A and HP 9830A/B Desktop Computers are used as the computing controllers in this manual.

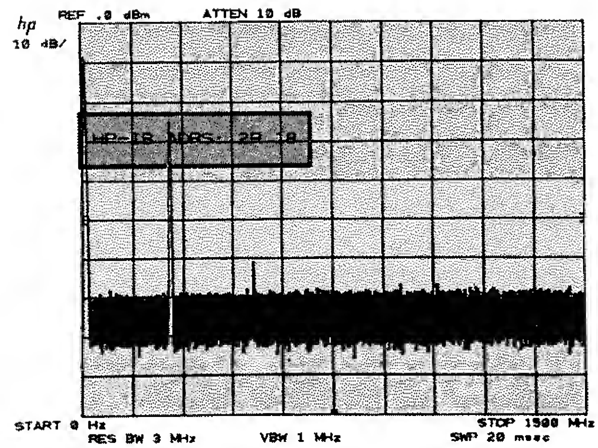
The following summarizes the required accessories for three computing controllers:

Computing Controller	Required HP-IB Interface	Required ROM Modules	Suggested ROM Modules	Language
9825A	98034A	GENERAL I/O	EXTENDED I/O	HPL
9830A		GENERAL I/O		
9830B	59405A	EXTENDED I/O		BASIC

The HP 8581A Automatic Spectrum Analyzer is an 8568A/9825A based system which includes a HP 9866B Printer, HP 98034A HP-IB Interface Card, computer cradle, system table, sample software programs and all the required accessories.

Addressing the Spectrum Analyzer

Communications between instruments on the HP-IB requires that addresses be assigned. The analyzer's address appears on the CRT display when the LINE power is turned from STANDBY to ON.



"HP-IB ADRS: 2 R 18"

LISTEN — 2

TALK — R

18 — read (TALK)/write (LISTEN) device code

Address Readout

Two formats are available for addressing an HP-IB instrument or device. One command format uses separate addresses for TALKING ("R") and LISTEN ("2"). The other uses only a device code ("18") to designate the recipient of the command. The latter format is used when the controller is the HP 9825A Computing Controller with the general I/O ROM.

Examples*

wrt 7 18, "<analyzer command>"

- 18 — 8568A device code, LISTEN
- 7 — HP-IB I/O Select code
- wrt — 9825A TALK

red 7 18, <controller variable>

- 18 — 8568A device code, TALK
- 7 — HP-IB I/O Select code
- red — 9825A LISTEN

CMD "? U 2", "<analyzer command>"

- 2 — 8568A LISTEN
- U — Controller TALK
- ? — unaddress all listeners (UNLISTEN)

CMD "? R 5", <controller variable>

- 5 — Controller LISTEN
- R — 8568A TALK
- ? — unaddress all listeners (UNLISTEN)

* In all examples, the preset addresses of the HP computing controllers are used:
9830A-9830B TALK "U"; LISTEN "5"; DEVICE CODE "21"
9825A HP-IB SELECT CODE "7"

REMOTE OPERATION

The read/write address of the 8568A can be changed from the front panel or the HP-IB by using the shift function P.

Pressing [SHIFT] [ZOOM] (P) [1] [8] [Hz/μV/μsec]

sets the address to 18. To set the address to 8

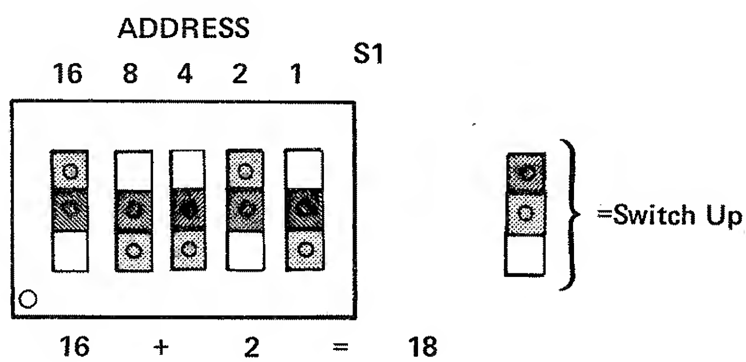
press [SHIFT] [ZOOM] (P) [8] [Hz/μV/μsec]

From the controller, the address can be set to 18 via the HP-IB using the 9825A statement

```
                         wrt 718,  "KSP   18   HZ"
code for [SHIFT] [ZOOM] (P) ──────┘     │     │
address ────────────────────────────────┘     └──── terminator
```

The switch on the analyzer's HP-IB interface card (switch S1 of A13) is used to determine what address will be used on line power on. The address is set as the sum of the numbers switched. For example, for the address 18, the 16 and 2 switches are pressed up and all others down.

The switch address 31 is a special code which commands the analyzer to use the last input address (whether from HP-IB or the front panel) upon line power on. If the address is lost*, the default address of 18 will be used.



HP-IB Address Switch on the Interface Card
(8568A Operating and Service Manual, Section II
describes the procedure for accessing the address switch.)

* The input address is stored in the analyzer's CMOS memory, which can remember for approximately 30 days with all line power disconnected.

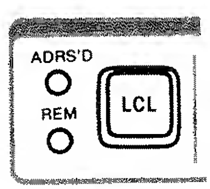
Setting the read/write address also sets the TALK/LISTEN ADDRESS. The following table lists the various combinations.

TALK	LISTEN	GENERAL ADDRESS
@	SP	00
A	!	01
B	"	02
C	#	03
D	$	04
E	%	05
F	&	06
G	'	07
H	(	08
I	)	09
J	*	10
K	+	11
L	,	12
M	–	13
N	.	14
O	/	15

TALK	LISTEN	GENERAL ADDRESS
P	0	16
Q	1	17
R	2	18
S	3	19
T	4	20
U	5	21
V	6	22
W	7	23
X	8	24
Y	9	25
Z	:	26
[	;	27
\	<	28
]	=	29
∧ or ⌒	>	30
Use last entered address at future power ON (set at factory)		31

Remote/Local Operation

If the controller has addressed the analyzer to TALK or LISTEN, the ADRS'D light will be on. When the analyzer is addressed with an HP-IB device command, the analyzer will go to remote, and the REM light will also go on.



Remote operation generally prevents front panel control of the analyzer except by those functions not programmable: LINE power, calibration and display adjustments, video trigger vernier and LCL. See Chapter 2.

Return to front panel, or local control by

pressing LCL

or executing a local device command from the controller such as

lcl 718

CAUTION

An operating HP-IB may be disrupted if the analyzer's LINE power is cycled. An analyzer should be connected to an **operating** HP-IB only with POWER ON.

Similar HP-IB disruption may result from pressing LCL when the HP-IB is active, thus a local lockout is recommended during 8568A automatic operation.

Analyzer Response to HP-IB System and Device Commands

HP-IB **system** commands effect all the instruments on the bus. HP-IB **device** commands affect only the device addressed. The spectrum analyzer response to each of the system and device commands is summarized on the next page. The following addresses are used:

	GENERAL ADDRESS	TALK	LISTEN
ANALYZER TALK/LISTEN ADDRESS	18	R	2
9830A/B ADDRESS	21	U	5

OUTPUT COMMANDS

HP-IB Message	Specific Controller Statements		Analyzer Response
	9825A	9830A	
Data Transfer	1:wrt 718,"<command>"*	10 CMD"?U2","<command>"	<listen> <go to remote>, executes command
System Trigger	<listen>,<go to remote> 1:trg 7	<listen>,<go to remote> 10 CMD "U" 20 FORMAT 3B 30 OUTPUT (13,20)256,8,512	<go to remote>, new sweep triggered
Device Trigger	1:trg 718	10 CMD "?U2" 20 FORMAT 3B 30 OUTPUT (13,20)256,8,512	
System Clear	<go to remote> 1:clr 7	<go to remote> 10 CMD "?U" 20 FORMAT 3B 30 OUTPUT(13,20)256,20,512	<instrument preset>
Device Clear**	1:clr 718	10 CMD "?U2" 20 FORMAT 3B 30 OUTPUT (13,20)256,4,512	<go to remote> <instrument preset>
System Remote Enable	1:rem 7	10 CMD "?U" 20 FORMAT B 30 OUTPUT (13,20)768	allows <go to remote> by setting HP-IB REN line true
Device Remote	1:rem 718	10 CMD "?U2" 20 FORMAT B 30 OUTPUT(13,20)768 40 CMD "2"	sets REN Line true <go to remote>
System Local	1:lcl 7	10 CMD "?U" 20 FORMAT B 30 OUTPUT(13,20)1024	<remote disabled> REN false, <go to local>
Device Local	1:lcl 718	10 CMD "?U2" 20 FORMAT 3B 30 OUTPUT(13,20)256,1,512	<go to local>
System Local Lockout	<remote enabled> <addressed> 1:llo 7	10 CMD "?U2" 20 FORMAT 3B 30 OUTPUT(13,20)256,17,512	no response to LCL
Clear Local Lockout and Set Local	1:lcl 7; rem 7 or press RESET	10 CMD "?U2" 20 FORMAT 2B 30 OUTPUT(13,20)1024,768	response to LCL
Abort	1:cli 7	Press STOP (will not go to local)	<unaddress> but if in local, will remain in local
Read Status Byte	1:rds (7)→A	10 CMD "2U" 20 FORMAT 5B 30 OUTPUT(13,20)256,95,53, 24,512; 40 CMD "R" 50 A = R BYTE 13 60 CMD "?U" 70 FORMAT 3B 80 OUTPUT(13,70)256,25,512;	See Chapter 5, Interrupt and Service Request Capability

* < > indicates an analyzer or controller executed command.

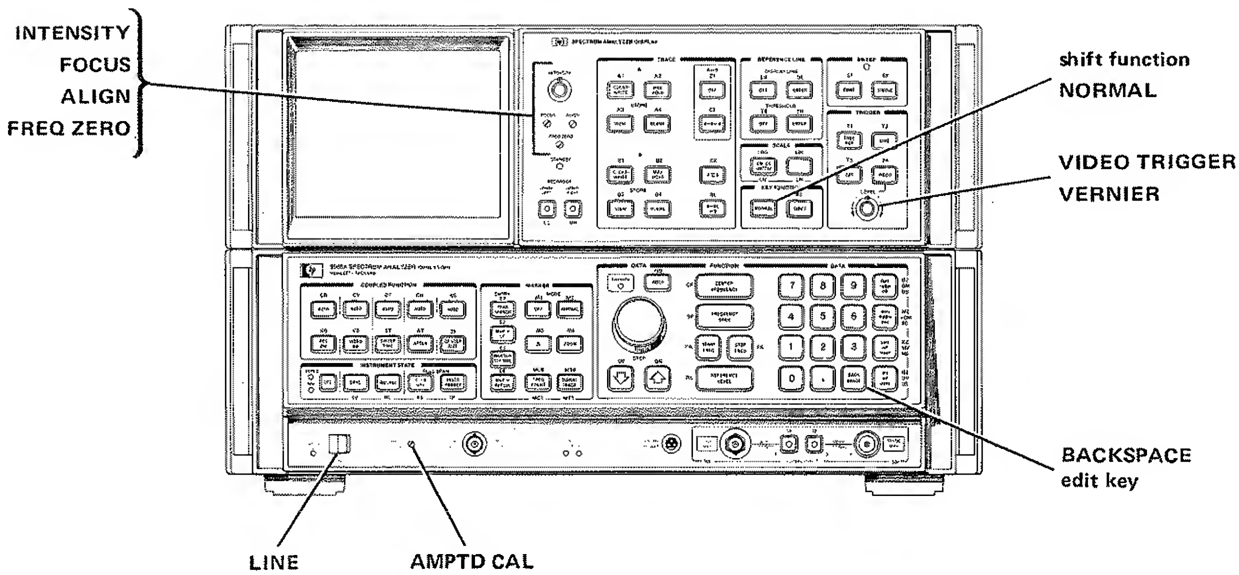
** A device clear can command an instrument preset even if the analyzer is locked in an unexecutable command sequence.

Chapter 2

HP-IB OPERATION OF FRONT PANEL CONTROLS

This chapter describes remote operation of the front panel controls, including the shift functions.

Since most of the controls can be remotely programmed by a controller, it is simpler to describe the controls **not** programmable.



Controls **Not Directly** Programmable

A full listing of those controls that can be remotely operated by a controller can be found with the instrument front panel drawing inside the rear cover.

Front Panel Control Commands

The analyzer responds to a remote front panel command the same way it does to a front panel command. In other words, the analyzer will behave the same whether the control changes come from the system controller with the analyzer in (remote) or the operator (with the analyzer in local).

PROGRAMMING HINT

When writing a program for operation of the analyzer's front panel controls, manually follow the measurement procedure and note the individual steps taken. The same chronological order of these steps can then be used to form the basis of the controller program.

Controller commands follow the same sequence as in manual operation:

1. activate function
2. change function value, if appropriate, with a data entry (including a terminator)

Functions are activated by a two or three character "function code"*. A data entry consists of numbers terminated with a units code. For example, to set the frequency span to 125.3 MHz

wrt 718, "SP 125.3 MZ"

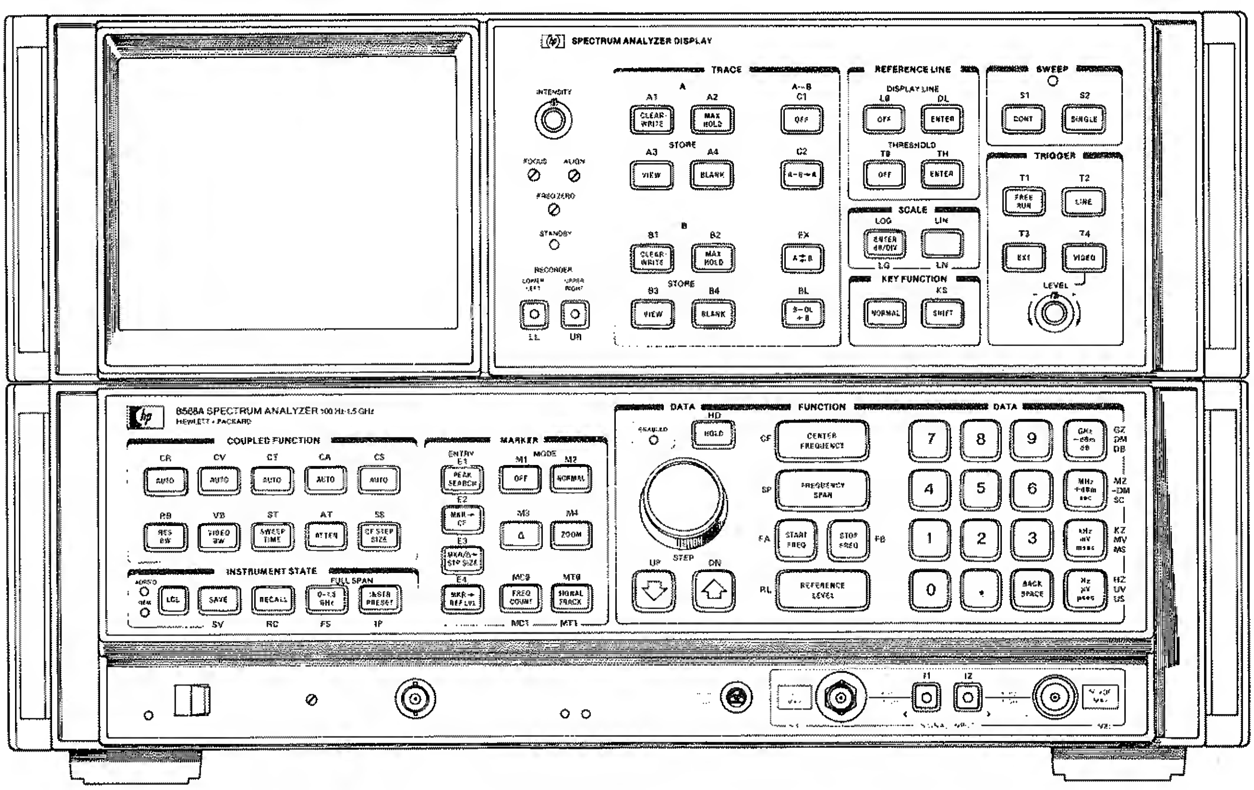
HP-IB Data Command — wrt 718,
frequency span function code — SP
data value — 125.3
entry terminated with MHz unit code — MZ

When the line is executed the analyzer responds to MZ just as it would to pressing [MHz -dBm sec] on the front panel.

Function Codes

A front panel function is activated by a two or three character code*. This code generally abbreviates the function name. For example, center frequency CF, reference level RL, and instrument preset IP. Some keys grouped together are numbered by location, such as the marker modes M1, M2, M3 and M4.

A complete summary is in a fold out inside the rear cover. This summary includes the preset conditions with instrument preset, [INSTR PRESET] or IP.



Front Panel Codes
(also see foldout inside rear cover)

* ASCII letters, numbers and symbols.

SHIFT KEY FUNCTION Codes

Programming a shift function requires a code sequence similar to the manual procedure for activating a shift function, that is, press SHIFT , then press the key with the function's code (the front panel blue character).

For example, to select the video averaging shift function, blue code G, execute

```
                        wrt 718,    "KS    G"
HP-IB Data Command ───────┘           │     │
activates shift ──────────────────────┘     │
activates video averaging ──────────────────┘
```

About half of the shift key function codes require ASCII lower case letters or symbols.

> **NOTE**
>
> Some controller keyboards do not offer lower case ASCII codes directly. For example, the HP 9830A Computing Controller keyboard outputs the ASCII upper case letters when unshifted. But when shifted the output letters are lower case, even though the 9830A display and companion HP 9866A Printer only show capital letters. To activate the shift function m, graticule off
>
> press 9830A (K) (S) (SHIFT) (M)
>
> The statement line appears as:
>
> 10 CMD "?UR", "KSM"
>
> The code to the analyzer will be sent as KSm. (KSn will turn the graticule on again.)

Data Entry

A data entry through the HP-IB must meet the same requirements as a front panel DATA entry, that is, it must have a number (value) and a message that terminates the entry, signaling the analyzer to assign the function value.

Number

The number code within the quote field must be a string of (ASCII) decimal numbers plus an optional decimal point. It may be preceded by a minus or plus sign. If the decimal is not included in the entry, it will be assumed at the end of the number. Either fixed or floating point notation may be used to make number entries. For example, the entries "12.3E6", "12.3e6" and "12300000" each will enter the same number. Caution should be exercised when using the "E" exponent format, since several marker command mnemonics also begin with E.

The number of significant digits accepted and stored by the analyzer is dependent upon which function is active. For example, an entry of 10 significant digits to center frequency will be stored in the analyzer's center frequency register.

If no number is entered, a "1" will be assumed.

Numbers as Variables

A data entry can be a controller variable as long as the format and individual controller statement syntax rules are followed.

For example, this program changes the center frequency in 100 MHz steps from 100 MHz to 1200 MHz:

```
0: fmt 1,f.0,c,z
1: for F=100 to 1200 by 100
2: wrt 718.1,"CF",F,"MZ";wait 2000 ◄── wait statement is for
3: next F                                viewing convenience
```

The variable F substitutes for the data entry number. The format "f.0,c,z" ensures that all digits of the variable F will be output from the controller with the leading spaces suppressed and no CR/LF's to prematurely terminate the entry. See FORMAT PRECAUTIONS below.

Terminating the Data Entry

The units code is the most common data entry terminator. It sets the value units and enters the function value.

Frequency	Code	Power	Code	Voltage	Code	Time	Code
Hz	HZ	dBm	DM	mV	MV	sec	SC
kHz	KZ	– dBm	– DM*	μV	UV	msec	MS
MHz	MZ					μsec	US
GHz	GZ	dB	DB				

Unit Codes

Other ASCII codes than the units codes can be used to terminate a data entry.

symbol	name	decimal equivalent (ASCII)
,	comma	44
CR	carriage return	13
LF	line feed	10
;	semi colon	59
ETX	end of text	3

ASCII Codes Which Terminate a Numeric Data Entry

These non-unit code terminators originate in the controller's language.

A terminated entry without a units code defaults to the fundamental units for the function activated. The default units of power depend upon the amplitude readout units selected.

Frequency	Hz
Power	± dBm, dBmV, dBμV or dB
Voltage	Volts
Time	seconds

Default Units

For example,

wrt 718, "CF 1200" < CR/LF > **

results in setting the center frequency to 1200 Hz.

* Either "–10.0 DM", "10.0–DM", or –10.0–DM" results in a negative entry.

** Execution of the HP 9825A Computing Controller wrt statement calls for a CR/LF as shown. < > indicates a transmitted message that does not appear as program text.

wrt 718, "KSB RL30" < CR/LF >

The reference level will be set to + 30 dBmV since those are the amplitude units selected by the code KSB.

If the unit used as a terminator is not correct for the activated function, the analyzer will select the unit that fits from the chosen unit's group. Unit groups are those written on the DATA keyboard's units keys.

GHz	MHz	kHz	Hz
+ dBm	− dBm	mV	μV
dB	sec	msec	μsec

Unit Groups

For example,

wrt 718, "RL 22 DB"

will enter + 22 dBm to the reference level.

FORMAT PRECAUTIONS

The controller's numeric **output** to the analyzer must be formatted such that the values are

1) not truncated;
2) not terminated by a CR/LF before the units code can be transferred.

Either fixed or floating point formats can be used, although the above conditions are more readily met in fixed point.

Generally the free field formats for computing controllers are sufficient to meet these requirements. However, it is best to retain explicit control of the controller format.

Format is one of the first items to check in a program which locks up the HP-IB or results in erroneous answers. Programs **without** explicit format statements may run or may not run, depending upon the controller's last format condition.

The free field format of the 9825A Computing Controller (equivalent to an output format of 4f18.4) will be sufficient for most single front panel commands. However when four or more commands are issued on one line, the fourth number will be terminated by a CR/LF prior to the units transfer. A better format is

0 : fmt **1** , **f.0** , **c** , **z**

- **1** — format number
- **f.0** — fixed point; suppresses leading spaces, and digits to the right of the decimal.
- **c** — character field for units mnemonic
- **z** — suppresses all CR/LFs

Example:

```
0: fmt 1,f.0,c,z
1: wrt 718.1,"CF",100,"MZSP",200,"KZRB",100,"KZRL",-20,"DM"
```

Line 1: format 1 is referenced in the write select code 718.**1**.

Note that reads (where data is input to the controller) are best transferred in the free field format. This is not the same format as for write. See 9825A General I/O Manual, page 8.

DATA STEPS ⇩, ⇧ Commands

The DATA step key's codes are

⇩ DN step down

⇧ UP step up

These codes can be used in programming just as the step keys are used for front panel data entry.

One useful application of the step key codes is for changing the values in a sequence, such as for center frequency step and frequency span.

For example, to decrease the resolution bandwidth by a factor of 100, four down steps are required (the resolution bandwidth sequence is 1,3,10).

In this example, the step command is used to change the center frequency by a preset step size:

wrt 718, "SS 150 MZ"	set CF step size to 150 MHz
wrt 718, "CF UP UP"	activates center frequency and increases center frequency by 2 steps of 150 MHz

DATA Knob Enable EK

The DATA knob can be activated for use from the front panel allowing the operator to select a specific value while under program control.

The code for activating the DATA knob

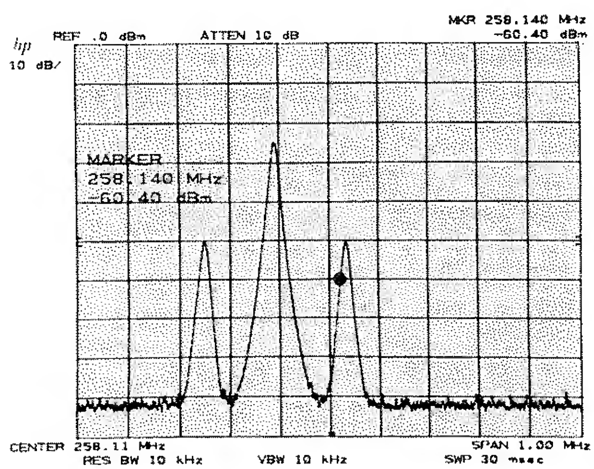
EK enable knob

For example, if the operator is required to set a marker on a signal response for further analysis by the controller, the following statement can be used:

wrt 718, "M2 EK" ; stp

- activates single marker (NORMAL mode) — M2
- enable DATA knob — EK
- stops the program — stp

At execution of this program line the operator moves the marker to the signal with the DATA knob, then continues the program with the 9825A CONTINUE



Only the DATA knob is enabled upon "EK". The other DATA controls and function keys are disabled. Enabled light is on.

DATA Keyboard Enable EE

Entry of data from the keyboard to the controller is possible with the use of the enable entry command "EE". A complete discussion of "EE" is in Chapter 3, Output Commands.

Learn Mode OL

The analyzer is able to output the current instrument state*, 80 bytes, through the HP-IB into controller memory. The controller can reestablish the instrument state by reading the 80 byte "learn string" back into the analyzer. The process is analogous to the front panel [SAVE] and [RECALL].

The command for the learn mode is:

OL output learn bytes

The learn mode requires:

1. uninterrupted transfer of all 80 bytes
2. controller ability to transfer 8 bit binary bytes (wtb and rdb statements in 9825A)

The most convenient form of controller memory for storage and recall of the learn bytes is an array.

To store and recall a single instrument state in an array such as A [ ]:

```
0:
1: dim A[80]
2: wrt 718,"OL"
3: for N=1 to 80
4: rdb(718)→A[N]
5: next N
6:
7:    instrument
8:    state changed
9:
10:
11: for N=1 to 80
12: wtb 718,A[N]
13: next N
```

```
10 DIM B[80]
20 FORMAT B
50 CMD "?U2","OL"
60 CMD "?R5"
70 FOR I=1 TO 80
80 B[I]=RBYTE13
90 NEXT I
100 CMD "?U2","IPTS"
110 ←——— instrument state changed
115 CMD "?U2"
120 FOR I=1 TO 80
130 OUTPUT (13,20)WBYTEB[I];
140 NEXT I
150 END
```

Note that the array A [ ] must be transferred with binary read and write statements.

Line 1 (10): Allocates controller memory for a single instrument state.

Line 2 (50): Initiates learn mode.

Lines 3-5 (60-90): Reads learn data from analyzer into controller variable A [ ].

Lines 11-13 (115-140): Writes learn data from A [ ] into the analyzer, reestablishing the original instrument state. The first input A [1] prepares the analyzer to accept the new instrument state.

When storing many instrument states with the learn mode, controller memory can be conserved by using a string array to store the values in integer precision.

```
16: dim A$[80]
17: wrt 718,"OL"
18: for I=1 to 80
19: char(rdb(718))→A$[I,I]
20: next I
21:
22:
23:
24:
25:
26:
27: wrt 718,A$
```

Line 16 to 20: saves the instrument state in the string array A$ [,].

Line 25: recall the instrument state from the same string.

* Instrument state does not include trace data, the states stored in save registers 1 through 7 or some shift function states.

See the HP 9825A, Computing Controller string variable programming extended I/O and advanced programming manuals for a complete explanation of these statements.

The controller write and read commands are the "binary" form. This suppresses trailing CR/LF, so they are not accepted as part of the learn string by either analyzer or controller. This assures that the correct 8 bits of each of the 80 learn bytes will be transferred.

The shift functions recorded with the learn mode include:

frequency offset
amplitude offset
video averaging (excluding the number of samples)
normal/auto triggering

Several diagnostic aids are also saved.

Take Sweep Command TS

The take sweep command, TS, insures that the analyzer will start and complete one full sweep before the next command is executed. Until a sweep is completed, the analyzer HP-IB will not respond to commands from the bus. One TS is required for each trace in the write mode.

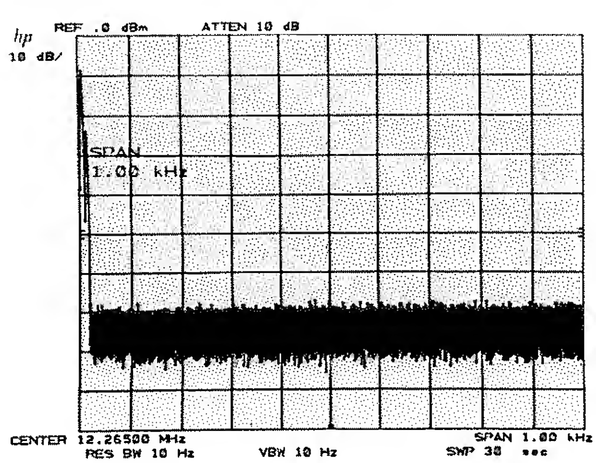
TS triggers a new sweep when the TRIGGER conditions are met.

The FUNCTION, MARKER, TRACE, COUPLED FUNCTION commands and a number of the shift functions require one complete sweep to update the display and trace memory. This is important for the output of measurement data either on the CRT display or through the HP-IB. (See Chapter 3 for output command use of TS.) For example, after a specific set of instructions, a viewed trace A is desired so the following is executed:

wrt 718, "IP CF 12.265 MZ SP 1000 Hz A3"

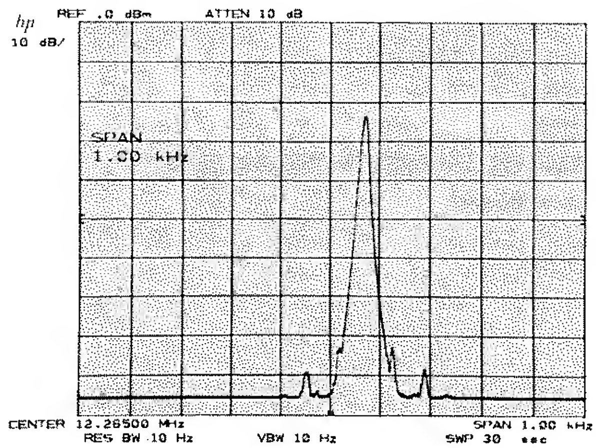
instrument preset — IP
center frequency — CF 12.265 MZ
frequency span — SP 1000 Hz
view in trace A — A3

This command sequence allows insufficient time for a full sweep between setting the span and activating the trace view mode, so only the full span, which was set by the instrument preset, is shown in the viewed trace A.



A take sweep command should be inserted before the view command calling for one complete sweep before execution of A3.

wrt 718, "IP CF 12.265MZ SP 1000 HZ TS A3"



Since the marker is repositioned at the end of each sweep when the marker is on, a TS guarantees that the marker will be on the trace current response before the analyzer TALKS. This is important for outputting the correct marker amplitude and frequency information through the HP-IB.

Service Request

When the analyzer is not able to interpret a command, a request for service is made. The CRT display will show "SRQ 140".

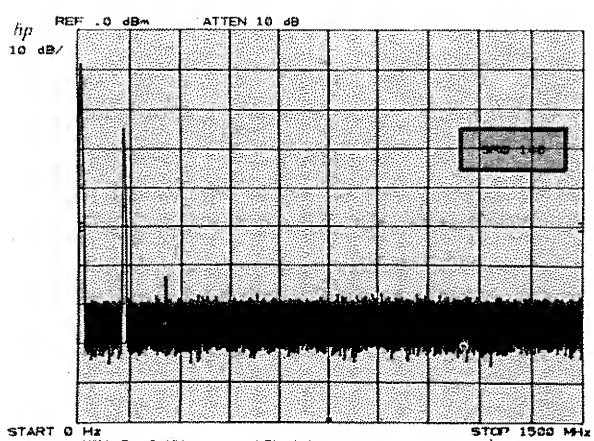
Further information on controller service request response (for interrupt) and interpretation (reading status byte) can be found in Chapter 5.

For example, the following statements result in "140" service requests:

wrt 718, "Cf 126 MZ" improper function code, should be "CF 126 MZ"

wrt 718, "CF 126 mZ" improper units code, should be "CF 126 MZ"

wrt 718, "CF, r1, MZ" improper syntax for variable r1, should be "CF," r1, "MZ"



Chapter 3

HP-IB OUTPUT COMMANDS

This chapter describes the commands which make possible the output of information from the analyzer through the HP-IB. See Chapter 1, 8568A Operation HP publication 08568-90002, for a discussion of the other rear panel outputs.

The types of information output through the HP-IB are summarized below:

Output	Output Command Codes	Pages
specific CRT display readout	OA, MF, MA	3.3
total trace information	TA, TB	3.6
specific display memory contents	DR	3. 9 (also see Appendix B).
instrument state learn mode	OL	Chapter 2
operator entered DATA values	EE OA	3.11
all display annotation	OT	3.5
operator enabled service request: SHIFT r	none	Chapter 5

Command Sequence

An output command tells the analyzer to TALK, and outputs the value on the HP-IB data lines. The controller must then LISTEN and prepare to store the value in memory. The following example illustrates:

```
2:
3:
4:
5: wrt 718,"O1 MA"
6: red 718,A
7:
8:
```

```
10 CMD "?U2","O1 MA"
20 CMD "?R5"
30 ENTER (13,*)A
40 END
```

Line 5 (10): Commands the analyzer to TALK, "wrt 718", and output in decimal, O1, the marker amplitude MA.

Line 6 (20,30): The controller is set to LISTEN and store the marker amplitude level in A.

Output Formats

Outputs to the controller through the HP-IB must be formatted appropriately for the controller and measurement requirements

Analyzer Output	Format Command Code	Output Examples
Decimal value in Hz, dB, volts or seconds	O3	0.52 (volts) 1072367 (Hz) −10.63 (dBm) .005 (sec)
Decimal values 0 to 4095: Representing trace amplitude CRT positions [1] 0 to 1023 positive and unblanked 2048 to 3071 positive and blanked 3072 to 4095 negative and blanked[2] Representing analyzer machine language words[3] (1024 to 4095)	O1	122 unblanked 1001 unblanked 2050 = +2, blanked 3995 = − 100 blanked 1056 machine language control word
Binary values in two 8 bit bytes, with the 4 most significant bits = 0. The most significant byte is output first.	O2	0000XXXX XXXXXXXX values 0 to 4095
Binary values in one 8 bit binary byte.[4] Amplitudes only.	O4	XXXXXXXX values 0 to 255

The binary formats are used primarily for the rapid transfer of function values which can be expressed in display units.

Value units format, O3 is selected on instrument preset.

1. Decimal values for frequency (x) and amplitude (y) are referenced from the lower left corner of the graticule (0,0). The values represent position in CRT display units. See Chapter 4 and Appendix A.
2. Negative values in the O1 output format are represented by the 12 bit two's complement of the negative number, that is, 4096 − \| negative value \|. For example, a −300 value would be output as 4096 − \| −300 \| = 3796.
3. Analyzer machine language programming is discussed in Appendix B.
4. The O4 output byte is composed of bits from the two bytes of the O2 output as follows:

```
0000XXXX XXXXXXXX      O2
    ||//////
    XXXXXXXX           O4
```

In O1 and O3 formats, only the exact number of characters to be output is transmitted, that is, a variable length string. Each item is ended by a CR/LF (ASCII 13 and 10). An HP-IB end or identify (EOI) accompanies the last LF.

Controller Formats

It is essential, when reading data from the analyzer, that the format of the controller be compatible with the output format of the analyzer.

Analyzer Format	Controller Format		
	Requirements	Examples	
		9825A	9830A
01	free field	0:fmt f	10 FORMAT
03	field size dependent on function output, use free field format	0:fmt	10 ENTER (13,*)
02	binary, read twice for each value	3:rdb 718,A 4:rdb 718,B	20 RBYTE (13)A 30 RBYTE (13)B
04	binary, read once for each value	0:rdb 718,A	30 RBYTE (13)A

Output of CRT Display Annotation OA MA MF OT

Any value that can 1) be read out on the CRT display and 2) be entered from the DATA controls can be output to the controller.

Three types of commands call for these outputs:

- Output the active function value OA
- Output the marker amplitude MA or the marker frequency MF
- Output the entire CRT readout as strings OT

Output Active Function OA

After a function is activated by either the operator or the controller, its present value can be output.

OA output active function value

The value unit format O3 is automatically selected with an output active function command.

COUPLED FUNCTIONS output using OA are left in MANUAL.

Examples best illustrate the use of OA.

The center frequency is stored in F and printed in the value unit, Hz.

```
0: fmt
1: wrt 718,"IP CF1234MZ"
2: wrt 718,"OA"
3: red 718,F
4: prt F
```

```
10 CMD "?U2","IP CF1234MZ"
20 CMD "?U2","OA"
30 CMD "?R5"
40 ENTER (13,*)F
50 PRINT F
60 END
```
*

When run, the 9825A printer outputs: 1234000000

*The spaces used within the quote field are for clarity and may be omitted. They will be ignored by the analyzer.

The sweep time, in seconds, is stored in T and printed.

```
0: fmt
1: wrt 718," ST50MS OA"
2: red 718,T
3: prt T
```

```
10 CMD "?U2"," ST50MS"
20 CMD "?U2","OA"
30 CMD "?R5"
40 ENTER (13,*)T
50 PRINT T
60 END
```

The 9825A printer outputs: 0.05

The video averaging sample size is stored in V and printed.

```
0: fmt
1: wrt 718,"IP KSG OA"
2: red 718,V
3: prt V
```

```
10 CMD "?U2","IP KSG OA"
20 CMD "?R5"
30 ENTER (13,*)V
40 PRINT V
50 END
```

The controller printer outputs the 100 sample limit size: 100

Marker Amplitude and Frequency Outputs MA MF

Whenever the markers are displayed, their amplitude and frequency (absolute or differential) can be output.

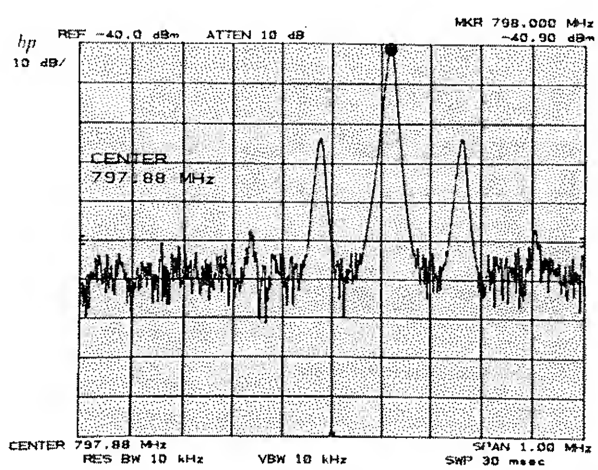
MF marker frequency output

MA marker amplitude output

The outputs can be formatted in amplitude and frequency units O3, decimal display units O1 or a binary output of display units, O2 or O4.

Unlike the OA command, the marker output commands do not require that the marker be activated at the time the output is made, only that the marker is being displayed on the CRT.

Example



Using the analyzer condition shown above, output the marker amplitude and frequency. Note that center frequency is the active function and not the marker.

```
0: fmt
1: wrt 718,"MF"
2: red 718,F
3: wrt 718,"MA"
4: red 718,A
5: prt F,A

 798000000.00    -40.90
```

```
10 CMD "?U2","MF"
20 CMD "?R5"
30 ENTER (13,*)F
40 CMD "?U2","MA"
50 CMD "?R5"
60 ENTER (13,*)A
70 PRINT F,A
80 END

798000000.00    -40.90
```

A read statement is used for each output command.

Serial **output** commands result in response only to the last command. For example, with

```
0: wrt 718,"MF MA"
1: red 718,F,A
```

reads only the marker amplitude into variable F. Marker frequency will not be read.

In zero span, MF will output the marker position in seconds.

Output All CRT Annotations OT

The output annotations command outputs 32 character strings, up to 64 characters long containing all the CRT readouts except readouts input with the label command LB.

OT Output annotation strings

To complete the command, all 32 strings must be read by the controller. The strings, in order of output, contain the following information:

String	Readout	String	Readout
1	"BATTERY"	17	frequency offset
2	"CORR'D"	18	video averaging
3	resolution bandwidth	19	title
4	video bandwidth	20	"YTO UNLOCK"
5	sweep time	21	"249 UNLOCK"
6	attenuation	22	"275 UNLOCK"
7	reference level	23	"OVEN COLD"
8	scale	24	"EXT REF"
9	trace detection	25	"VTO UNCAL"
10	center frequency or start frequency	26	"YTO ERROR"
11	span or stop frequency	27	"MEAS UNCAL", "*"
12	reference level offset	28	frequency diagnostics
13	display line	29	"2ND LO ↑", 2ND LO↓"
14	threshold	30	"SRQ"
15	marker frequency	31	center frequency "STEP"
16	marker amplitude	32	active function

OUTPUT COMMANDS TA TB

The following program stores all the readouts in the A$ string array:

```
1: dim A$[32,64]
2: wrt 718,"OT"
3: for N=1 to 32
4: red 718,A$[N]
5: next N
```

```
10 DIM A$[64]
20 CMD "?U2","OT"
30 CMD "?R5"
40 FOR N=1 TO 32
50 FORMAT B
60 ENTER (13,50)A$
70 NEXT N
```

line 1 (10): dimensions 32 strings each up to 64 characters long.

line 2 (20): the output command

lines 3 to 5 (30 to 70): reads each annotation message and stores it into a string. Strings without messages are null strings

After a LINE power ON, OT and a print routine outputs the following string arrays:

String Number	Contents	String Number	Contents
1			
2		17	
3	RES BW 3 MHz	18	
4	VBW 1 MHz	19	
5	SWP 20 msec	20	
6	ATTEN 10 dB	21	
7	REF .0 dBm	22	
8	10 dB/	23	
9		24	
10	START 0 Hz	25	
11	STOP 1500 MHz	26	
12		27	
13		28	
14		29	
15		30	
16		31	
		32	HP-IB ADRS: 2R 18

where all blank strings are null (empty).

Output of Trace Data TA TB

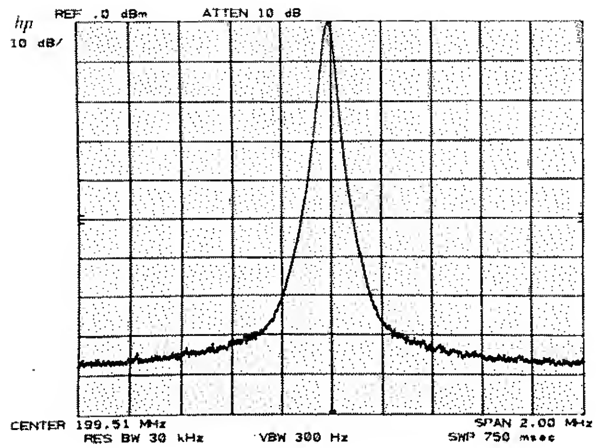
The CRT display traces are stored in digital memory as display unit values. (See Appendix A, Display Memory Structure). These values are output in left to right CRT sequential order by the commands.

TA output trace A

TB output trace B

Even though the trace amplitudes are stored in the display memory as display unit amplitude, they can be output in any of the four output formats.

Example



To store the above trace in the array A [1001] the following program would be run.

```
0: fmt
1: dim A[1001]
2: wrt 718,"O1 TA"
3: for N=1 to 1001
4: red 718,A[N]
5: next N
```

Line 1: set up storage for 1001 trace points

Line 2: sets format and commands trace A output

Lines 3-5: sequentially reads all 1001 trace points into A [N].

A printout of every one hundredth point reads in display units:

```
130.00
128.00
122.00
123.00
138.00
974.00
144.00
115.00
119.00
100.00
127.00
```

trace amplitude in display units

Changing the output format to O3, and rerunning the same program will change the array to read in units of power:

```
-87.90
-88.10
-88.70
-88.60
-87.10
 -3.50
-86.50
-89.40
-89.00
-90.90
-88.20
```

trace amplitude in dBm

To reduce the controller trace storage requirements only a fraction of the points need be stored. For example:

```
0: fmt
1: dim B[100]
2: fmt
3: wrt 718,"O1 TB"
4: for N=1 to 1001
5: red 718,B
6: if frc(N/10)=0;B→B[N/10]
7: next N
```

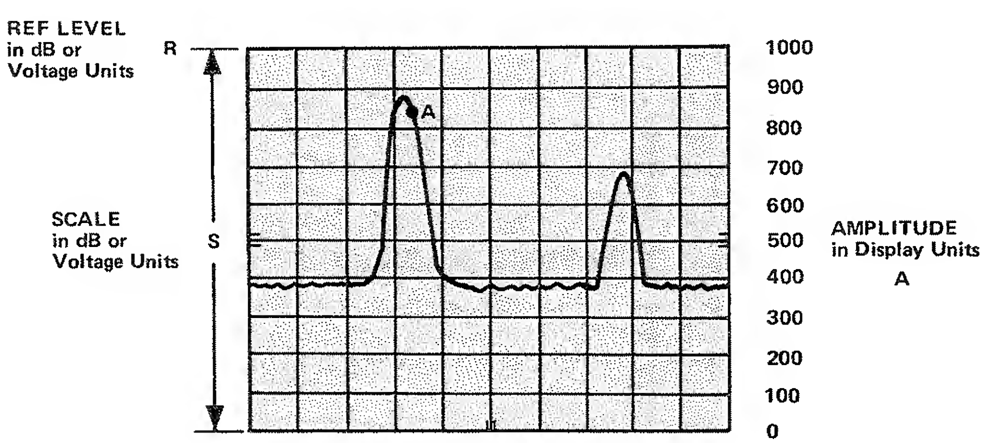
```
10 DIM B[100]
20 CMD "?U2","O1,TB"
25 CMD "?R5"
30 FOR N=1 TO 1001
50 ENTER (13,*)B
60 IF INT(N/10)=N/10 THEN 80
70 GOTO 90
80 B[N/10]=B
90 NEXT N
```

Line 5 (50): reads every point, but stores only every tenth value.

When transferring less than 1001 points of trace data, the output mode is most efficiently terminated with the DATA HOLD command HD.

Output of Trace Arithmetic Values

A – B→A and B – DL→B result in new trace values being placed in memory. When the result of either function lies above the bottom graticule the value is as read out from the display. When the function results in a trace that lies below the bottom graticule the trace will not be displayed. Calculation of the outputs are as follows:



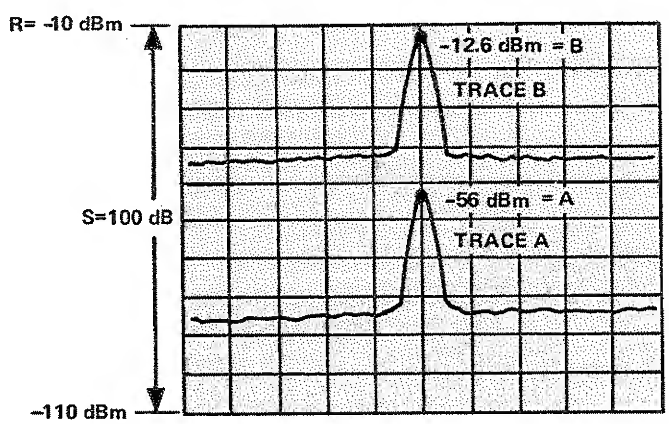
Function	Condition	Output Format		
		01 Display Units	03 LOG SCALE, dB/	03 LINEAR SCALE, Voltage
A – B – A	A > B A < B	A – B 4096 – (B-A)	A – B + (R-S)	A – B
B – DL – A	B > D B < D	B – D 4096 – (D-B)	B – D + (R-S)	B – D

Trace Arithmetic Output Values

where A is trace A position
B is trace B position
D is display line position
(R-S) is the absolute power level of the bottom graticule

Example

The following trace peaks are output:



After A – B – A, trace A blanks. If the peak of trace A is output, the value is

$$A - B + (R\text{-}S) = -56 - (-12.6) + (-10 - 100)$$
$$= -153.4 \text{ dBm in 03 format}$$

Reading Trace Amplitude at a Specific Location DR

The display read command allows the output of individual trace amplitudes, rather than all 1001 points.

DR display read

For output of trace information, the specific trace memory address must first be input to the analyzer with a display address command, DA. Appendix A, Display Memory Structure, explains memory addressing.

These program lines will read out the amplitude at the center frequency of trace A, trace B and trace C into A, B and C respectively.

```
1: fmt
2: wrt 718,"DA501 DR"
3: red 718,A
4: wrt 718,"DA1525 DR"
5: red 718,B
6: wrt 718,"DA3573 DR"
7: red 718,C
```

```
10 CMD "?U2","DA501,DR"
20 CMD "?R5"
30 ENTER (13,*)A
40 CMD "?U2","DA1525,DR"
50 CMD "?R5"
60 ENTER (13,*)B
70 CMD "?U2","DA3573,DR"
80 CMD "?R5"
90 ENTER (13,*)C
```

Learn Mode OL

Learn mode provides one means for saving the analyzer's instrument state, for later recall. A complete discussion of this command is in Chapter 1.

Operator Entered DATA Values EE

The controller can receive values entered by the operator from the analyzer's DATA keyboard using the enable entry command

EE OA enable and output from DATA keyboard and output

The general sequence of programmed events should be as follows:

1. A controller program loop is formed to prevent the controller from using the entered value until the operator signals that the entry is complete.
2. The operator makes a DATA entry, and signals completion of entry.
3. The controller reads the value of the entry and continues to the next program step.

Two methods can be used to exit the program loop depending on the type of output required.

output value	method
a single positive digit from 1 to 9	test the entry for non zero value
a positive integer from 0 to 999999999999	service request

Examples

Single digit entries can be made with the program.

```
1: fmt
2: wrt 718,"EE OA"
3: red 718,N
4: if N=0;gto -2
5: prt N
```

```
10 CMD "?U2","EE"
20 CMD "?U2","OA"
30 CMD "?R5"
40 ENTER (13,*)N
50 IF N=0 THEN 10
60 PRINT N
80 END
```

Line 1 (40): The readout of the entry should be in the free field format for the line 3 read statement.

Lines 2 to 4 (10 to 60): A program loop, exited only when a non-zero entry is made. The loop works this way: The enable entry command sets the entry to zero (default value). OA outputs the value, and the controller reads into N. If N = 0, as it will without a DATA entry, the program continues at line 2.

Line 5 (60): Outputs the entered number on the controller's printer.

The printed outputs from pressing various DATA keys is as follows:

1	1.00	GHz +dBm dB	1000000000.00
5	5.00	MHz −dBm sec	1000000.00
9	9.00	kHz mV msec	1000.00
		Hz µV µsec	1.00

There is no response to pressing DATA 0.

Multiple digit entries, and zero, can be made when the service request is used to exit a program loop.

```
1: fmt
2: wrt 718,"R1 R4 EE"
3: rds(718)→A;if bit(1,A)=0;jmp 0
4: wrt 718,"OA"
5: red 718,N
6: prt N
```

Line 1: The readout of the entry should be in the free field format (line 5 read statement).

Line 2: Enable entry command preceded by service request format statements. R1 clears the service request capability of the analyzer. R4 calls for a service request if a unit's key is pressed to signify the completion of an entry. See Chapter 5.

Line 3: A one line program loop which monitors the HP-IB service request status byte. When the unit's key is pressed a service request sets bit 1 of A true and the program continues to line 4. rds (718)→A clears the service request. See Chapter 5.

Line 4 and 5: The active function is output and read.

Line 6: Outputs the entered number on the controller's printer.

Now, positive integer values up to $10^{12}-1$ can be entered. This was illustrated by the following execution of the program.

DATA Entry	Output
1 Hz µV µsec	1.00
1 2 3 . 4 5 kHz mV msec	123450.00
1 2 3 . 4 5 Hz µV µsec	123.00

Operator Enabled Service Request

When the analyzer is operated from the front panel, an HP-IB controller can be programmed to respond to the service request. [SHIFT] r.

This allows the analyzer operator to call up a controller, and command its attention from the analyzer front panel, even if the analyzer was previously unaddressed in the HP-IB system. Chapter 5 discusses this technique in detail.

Chapter 4

HP-IB INPUT TO THE DISPLAY

This chapter discusses the commands which allow custom CRT display annotation and graphics.

The CRT display memories can store graphic and label data that are input by the HP-IB controller. For example, the display may be used to show the test setup block diagram, test data in a table or graph, instructions, or test limits drawn over the graticule. This data can be displayed on the CRT alone, or with the normal trace and annotation information.

	Command	Command Code
Drawing Vectors	plot absolute plot relative	PA PR
Words, characters and numbers	label	LB
Graphing functions	graph	GR
Inputting a trace	input trace B (in binary)	IB

Normally memory allocated to trace C is used to store and display HP-IB display input data so that trace A and B can be used in a normal manner. However, any part of memory may be used for graphics if required. Appendix B describes the necessary commands.

Clearing the Display

The CRT display can be cleared of annotation and graticule.

"IP A4 KSm KSo" clears display

Instrument preset erases the last graphics in trace C, blanks trace B, and assures that any new labeling or plotting will start in trace C. To erase trace C graphics and prepare the new trace C labeling or plotting without an instrument preset, use the command

"EM A4 B4 KSm KSo L0" clears display without instrument preset

"EM" erases trace C and allows new trace C input

"L0" clears the display line, if enabled.

NOTE

Trace C page is used for the execution of several shift functions. It is not possible to use the trace C page for special graphics *and* use the following shift functions:

KSG video averaging

KSw display error correction data

Sweeping Without a Trace Display

In the above clearing procedure A4, B4 blanks the A and B traces and stops the sweep. In order to continue sweeping and **not** display the trace information the following codes can be written to the display memory.

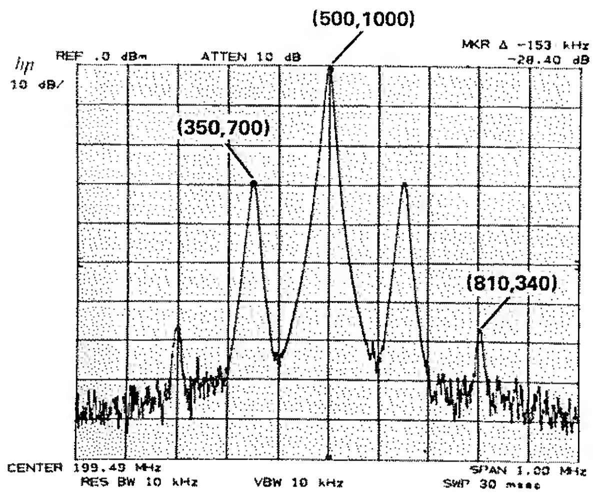
Trace A "DA0 DW 1056" or "DA0PS"

Trace B "DA1024DW1056" or "DA1024PS"

Appendix B discusses the DA, DW and PS commands.

CRT Display Reference Coordinates

Positions on the CRT display are referenced in display units. For example, the coordinates of several points along a trace can be designated as follows:

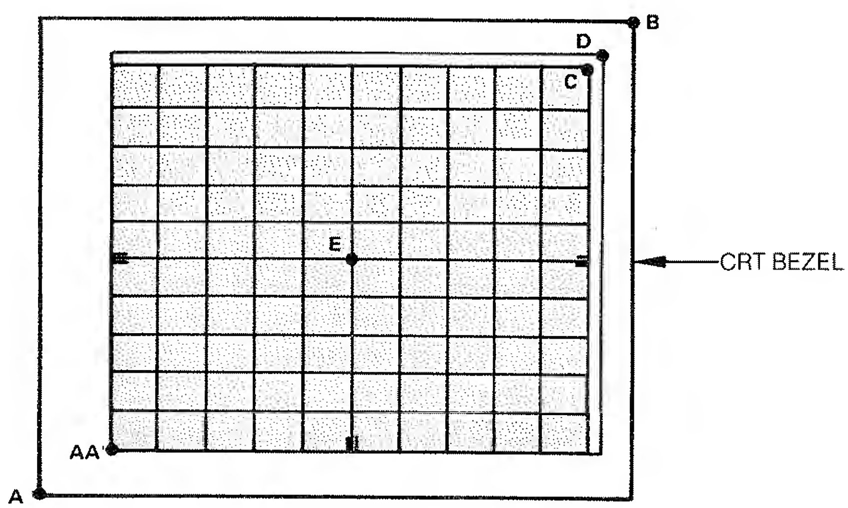


The numbers represent distance from the lower left hand corner of the graticule (X = 0, Y = 0) in display units. The upper right hand graticule corner is the (1000,1000) point. Also see Appendix A.

Three scales referenced to the CRT screen can be used for positioning on the CRT display. Each is initiated by a display size command code D1, D2, or D3. Once a code is selected it remains in effect until changed.

Preset Display Size D1, Full Screen Display Size D2, Expanded Display Size D3

The preset size is used for all graticule trace information. It is automatically called upon instrument preset.



Size	(0,0)	AA	B	C	D	E
D1	AA	(0,0)	*	(1000,1000)	(1023,1023)	(500,500)
D2	A	(120,73)	(1023,1023)	(1005,957)	—	(562,515)
D3	A	(81,49)	(689,689)	(676,645)	—	—

*No writing outside boundary marked by AA, D.

NOTE

The display size boundaries shown assume the CRT display has been internally adjusted to the standards set in the 8568A Operating and Service Manual.

Aspect Ratio

The aspect ratio of the display, that is, the proportions of x to y that will yield the same length on the CRT, is 1:1.3. For example to create a square figure on the display if the x vector were 100 units, the y vector would have to be 100 x 1.3 = 130 units.

Plot Commands PA PR

Graphics are plotted on the CRT display using vectors: lines whose end points are specified in display units. The vector coordinates can be given in absolute units (plot absolute, PA) or in units relative to the last plotted point (plot relative, PR).

PA plot absolute
PR plot relative

The pen commands, pen up PU and pen down PD, determine whether or not the vector specified is visible on the CRT.

PU pen up, next vectors are not displayed
PD pen down, next vectors are displayed

Notation Conventions

PA	Blue characters are command literals (ASCII code mnemonics).
[]	Items within brackets are optional.
. . .	Three dots indicate that the previous item(s) can be duplicated.
< >	Items enclosed in the angular brackets are considered to be elements of the language.
|	Indicates a choice of one element from a list.

Syntax for Plot Absolute

[PU|PD] PA x_1,y_1, [PU|PD] x_2,y_2, [PU|PD] x_3,y_3, . . .

The plot absolute command is followed by sets of x-y pairs. Each pair specifies a location relative to the display origin (0,0) specified in display units. Pen up (PU) and pen down (PD) dictate whether or not the following vector(s) are displayed.

Syntax for Plot Relative

[PU|PD] PR x_1,y_1, [PU|PD] x_2,y_2, [PU|PD] x_3, y_3, . . .

The plot relative command is followed by sets of x-y pairs. Each pair specifies a location relative to the last point specified in display units. Pen commands are used the same as in plot absolute.

> **NOTE**
>
> Each number entry, whether a literal or a variable, requires a delimiter to insure entry. The x and y values in the syntax statements must be delimited by a comma (,), semicolon (;), CR, LF or an ETX. The y value may also be delimited by a following command.

Example

The following programs draw a rectangular figure on the CRT.

```
0: fmt
1: wrt 718,"IP A4 KSm KSo"
2: wrt 718,"D2 PU"
3: wrt 718,"PA 700,500 PD 900,500"
4: wrt 718,"900,300,700,300,700,500"
5: end
```

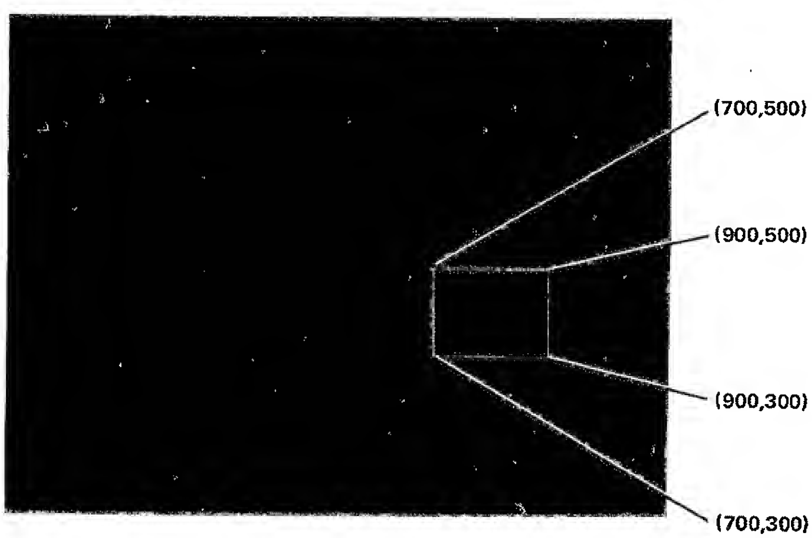
```
10 CMD "?U2","IP,A4,KSM,KSO"
20 OUTPUT (13,*)"D2,PU"
30 OUTPUT (13,*)"PA700,500,PD,900,500"
40 OUTPUT (13,*)"900,300,700,300,700,500"
50 END
```

Line 1(10): Instrument preset and clear the display.

Line 2(20): Specifies the full CRT display size. The pen up command insures that the initial vector to point (700,500) is not drawn.

Line 3(30): Plot absolute command and the starting point of the rectangle. The following pen down command assures that the vector (700,500) to (900,500) will be drawn on the CRT.

Line 4(40): Plots the remainder of the rectangle on the CRT. The pen down condition is still in effect.



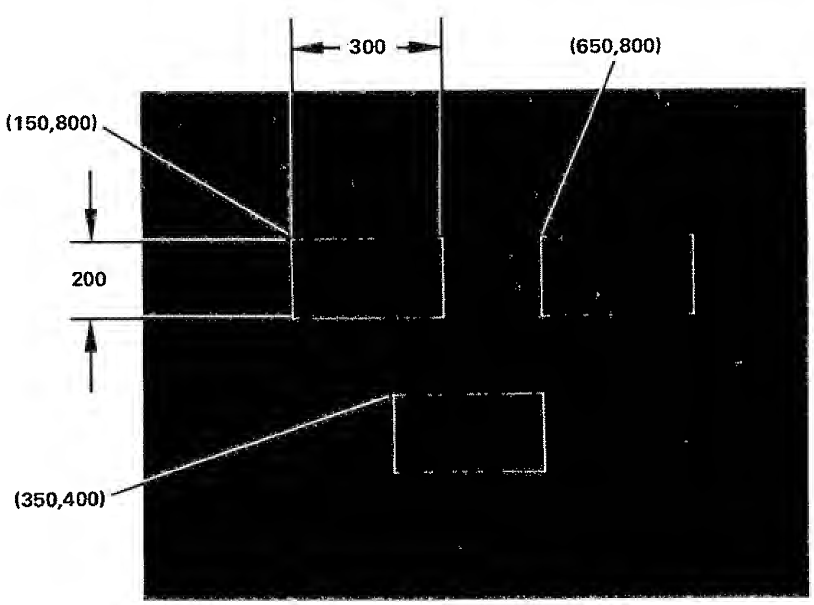
A similar rectangle can be positioned anywhere on the display using the plot relative command. The following programs draw the same figure in the three places on the CRT:

```
0: fmt
1: wrt 718,"IP A4 KSm KSo"
2: wrt 718,"PU PA 150,800"
3: gsb "rectangle"
4: wrt 718,"PU PA 650,800"
5: gsb "rectangle"
6: wrt 718,"PU PA 350,400"
7: gsb "rectangle"
8: end
9: "rectangle":
10: wrt 718,"PD PR 300,0,0,-200,-300,0,0,200"
11: ret
```

```
10 CMD "?U2","IP A4 KSM KSO"
20 CMD "?U2","PU PA 150,800"
30 GOSUB 200
40 CMD "?U2","PU PA 650,800"
50 GOSUB 200
60 CMD "?U2","PU PA 350,400"
70 GOSUB 200
80 END
200 REM RECTANGLE
210 CMD "?U2","PD PR 300,0,0,-200,-300,0,0,200,"
220 RETURN
```

Lines 2,4,6 (20,40,60): Position vector for the drawing of the rectangle. The vector is not displayed.

Line 10 (210): The plot relative command draws a rectangle 200 by 300 display units. Each pair of numbers inputs increments in the x-y pair.



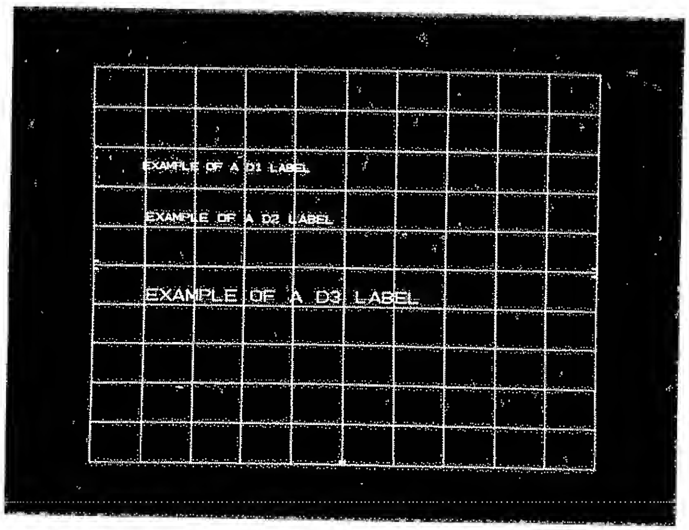
Label Command LB

The label command allows writing text anywhere on the CRT display.

LB   label

All HP-IB data messages following a label command will be written on the CRT display. Plot commands are used to position the text on the CRT.

The size of the label characters is determined by the display size.



Syntax for Label

<display size> [plot] LB <text> <label terminator>

LB	enables the label mode
<text>	a string of character codes (see below)
<label terminator>	a message which ends the label mode. The next ASCII character code will not be written on the CRT display.

<label terminator> can be done with either an ASCII ETX, end of text (decimal code 3) or a character code selected by the user with DT label terminator. For example, if the ? symbol is desired to end a label mode, execute the following statement:

wrt 718, "DT?"

An instrument preset will eliminate the special <label terminator> code, ?.

Example

In the following program, line 7 ends the label mode

6:wrt 718, "PU PA 500, 500 LB LABEL"
7:wtb 718, 3

If a ? is to be used to terminate the label mode this line is used before the label statement,

wrt 718 "DT?"

Now line 6 can be written

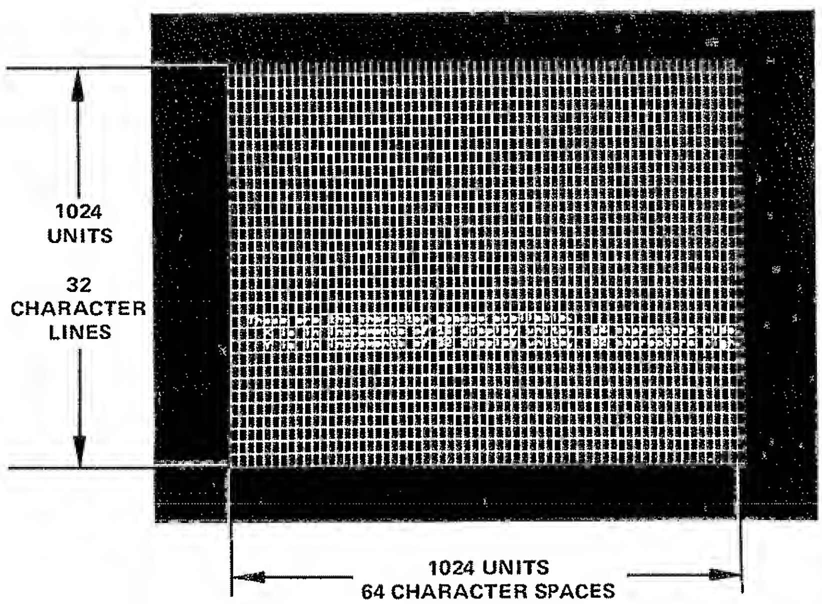
6:wrt 718, "PU PA 500, 500 LB LABEL?"

The ? terminates the label mode, and line 7 can be omitted. The terminator is not displayed, nor is it stored in display memory.

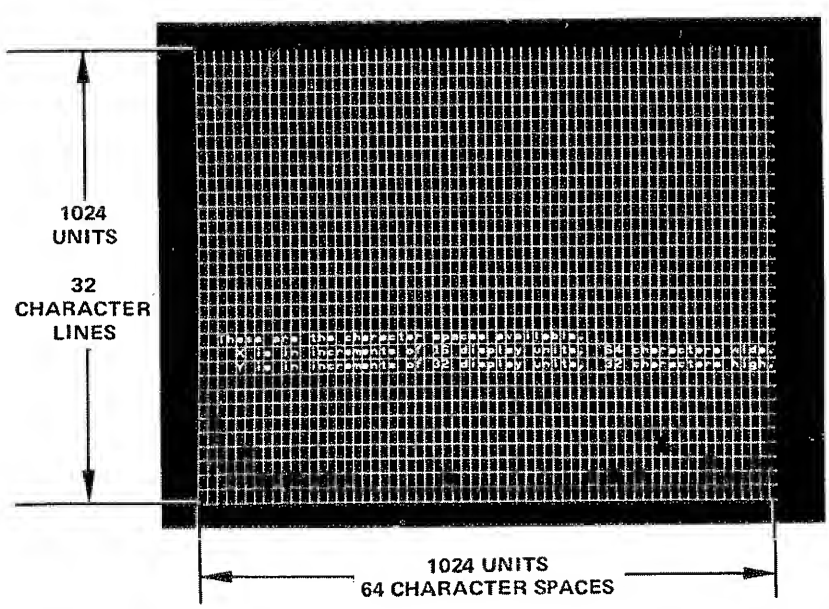
Character Position

Characters generated for the label command are placed on the CRT display in a fixed grid similar to the character positions on a typed page, that is, in rows and columns. This can be an important consideration when labeling graph lines or points. The display size determines position and number of rows and columns.

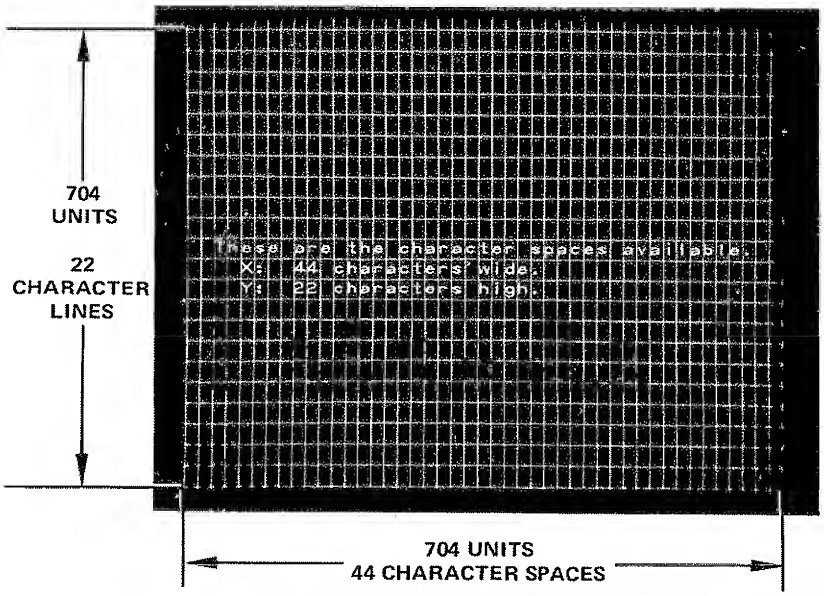
D1:



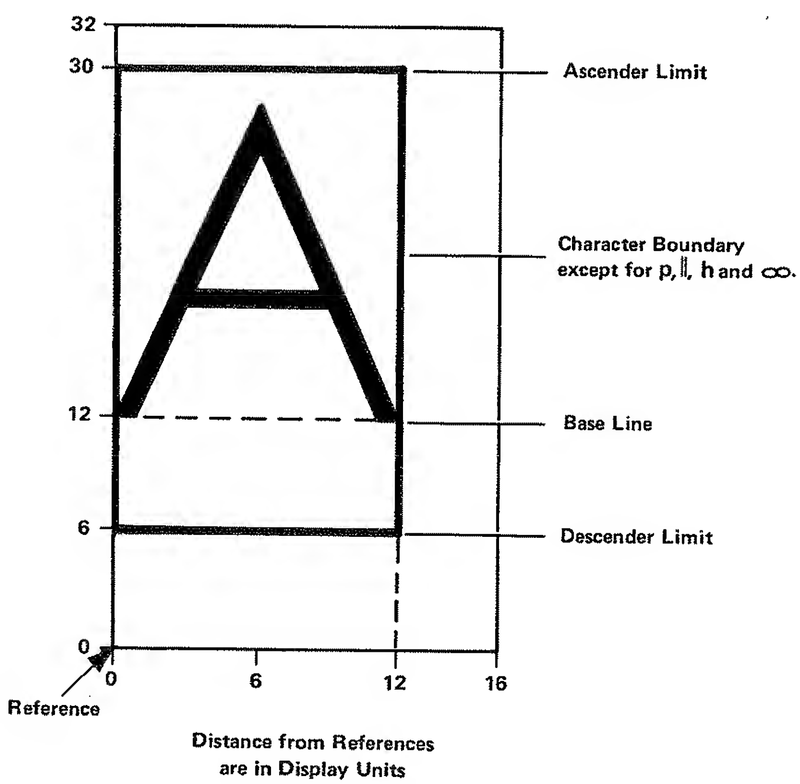
D2:



D3:



The character position is referenced from the lower left corner of the character space shown below. If a plot absolute statement calls a position anywhere within the space, the character will be placed within the "character boundary" in the space. If two characters are labelled into the character space, they will be superimposed over one another.



Single Character Space

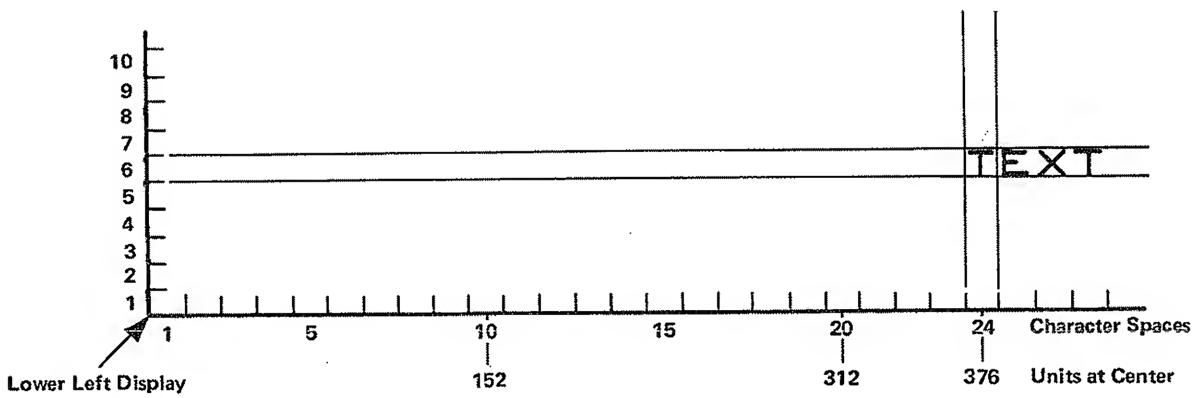
Example

To begin labeling text six characters up from the bottom and 24 characters from the left (in **any** display size), the plot absolute vector values are calculated.

$$x = (\text{character spaces})(16) - 8$$
$$= (24)(16) - 8 = 376$$
$$y = (\text{character spaces})(32) - 16$$
$$= (6)(32) - 16 = 176$$

" PU PA 376, 176 LB <text> "

The first character of text will be positioned as shown:



x may be changed as much as ± 7 units and y as much as ± 15 units before the text will begin at the next x and y character. In other words, the label positioning statements "PA 376, 176 LB <text>" and "PA383, 191 LB <text>" will place the text in the same character spaces.

Character Set

The character set for the label command is the same as the ASCII set. There are 86 additional characters available.

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
0	(NULL)	32	SP	64	@	96	`	128		160	∧	192	∠	224	ψ
1		33	!	65	A	97	a	129		161	ˇ	193	^	225	α
2		34	"	66	B	98	b	130		162	¨	194	∫	226	β
3		35	#	67	C	99	c	131		163	≠	195	¢	227	χ
4		36	$	68	D	100	d	132		164	£	196	∇	228	δ
5		37	%	69	E	101	e	133		165	∝	197		229	ε
6		38	&	70	F	102	f	134		166	⊕	198		230	φ
7		39	'	71	G	103	g	135		167	·	199	g	231	γ
8	(BS)	40	(	72	H	104	h	136		168	←	200	ℏ	232	η
9		41	)	73	I	105	i	137		169	→	201	ı	233	ι
10	(LF)	42	*	74	J	106	j	138		170	§	202		234	ζ
11	(VT)	43	+	75	K	107	k	139		171	±	203		235	κ
12	(FMFD)	44	,	76	L	108	l	140		172	↓	204	L	236	λ
13	(CR)	45	—	77	M	109	m	141		173	—	205	∩	237	μ
14		46	.	78	N	110	n	142		174	×	206	∩	238	ν
15		47	/	79	O	111	o	143		175	÷	207	o	239	ο
16		48	0	80	P	112	p	144		176	o	208	ρ	240	π
17	(BKON)	49	1	81	Q	113	q	145	(SK16)	177	ı	209	∞	241	θ
18	(BKOF)	50	2	82	R	114	r	146	(SK32)	178	≥	210	r	242	ρ
19		51	3	83	S	115	s	147	(SK64)	179	∃	211	s	243	σ
20		52	4	84	T	116	t	148		180	⊣	212	T	244	τ
21		53	5	85	U	117	u	149		181	≥	213	∂	245	υ
22		54	6	86	V	118	v	150		182	∃	214	v	246	ξ
23		55	7	87	W	119	w	151		183	√	215		247	ω
24		56	8	88	X	120	x	152		184	~	216	‾ _	248	Γ
25		57	9	89	Y	121	y	153		185	≃	217	ı ı	249	Δ
26		58	:	90	Z	122	z	154		186	‖	218		250	Ω
27		59	;	91	[	123	{	155		187	∴	219	Π	251	Σ
28		60	<	92	\	124	|	156		188	≤	220	Θ	252	Λ
29		61	=	93	]	125	}	157		189	≡	221	Ψ	253	ϒ
30		62	>	94	↑	126	~	158		190	≥	222	Φ	254	Ξ
31		63	?	95	_	127		159		191	^	223		255	

Label Command Character Set

Blank codes are either unassigned or character pieces. ( ) indicates display machine language word see Appendix B.

Examples of Label

In the following program the text is positioned on the CRT by a plot abolute command.

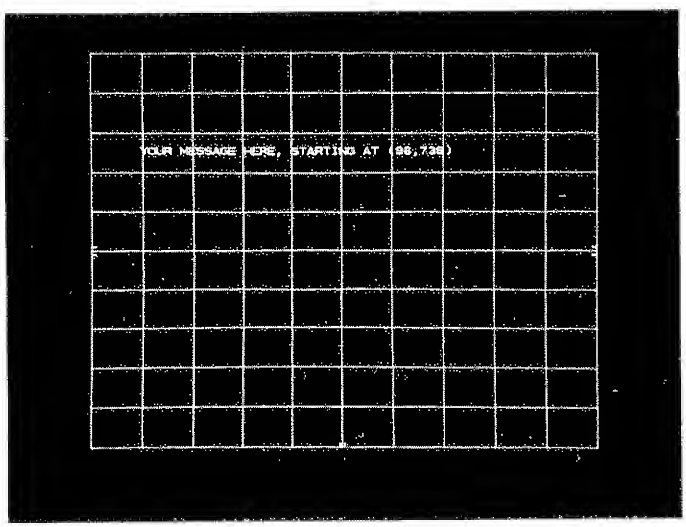
```
0: fmt
1: wrt 718,"IP A4 KSo"
2: wrt 718,"PU PA96,736 LBYOUR MESSAGE HERE,
                                        STARTING AT (96,736)"
3: wtb 718,3

10 CMD "?U2","IP A4 KSO"
20 OUTPUT (13,*)"PU PA 96,736 LBYOUR MESSAGE HERE,
                                        STARTING AT (96,736)"
30 FORMAT B
40 OUTPUT (13,30)3
50 END
```

Line 1(10): Prepares the display and selects display size D1.

Line 2(20): Positions the beginning of the text, enables the label mode and outputs the text.

Line 3(30,40): Ends the label mode.



Here is what happens if the <label terminator> message is omitted:

```
0: fmt
1: wrt 718,"IP,A4,KSo"
2: wrt 718,"PUPA0,500LBThis is the message you want."
3: for N=1 to 10
4: wrt 718,"CF100MZ,SP200KZ,ST50MS"
5: next N
```

Line 2: Since no <label terminator> ends the label mode the following program lines are also written on the CRT.

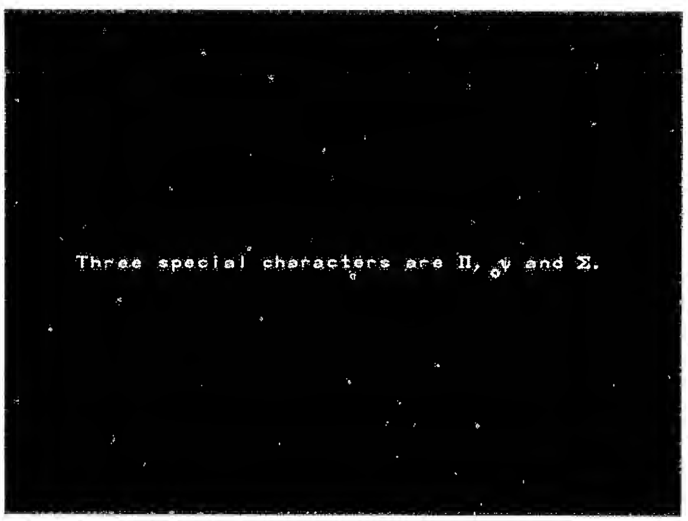
Lines 3 to 5: Any type of program

This final example shows how to use the special character set, and the DT terminator.

```
0:  fmt
1:  wrt 718,"IP A4 KSm KSo D3 DTz"
2:  wrt 718,"PU PA 25,350"
3:  wtb 718,"LBThree special characters are "
4:  wtb 718,219,44,32,224,32,"and ",251,".z"

10 CMD "?U2","IP A4 KSM KSO D3 DTZ"
20 OUTPUT (13,*)"PU PA25 350"
30 FORMAT B
35 FORMAT 7B
40 OUTPUT (13,30)"LBTHREE SPECIAL CHARACTERS ARE ";
50 OUTPUT (13,35)219,44,32,224,32,"AND ",251,".Z"
60 END
```

Line 4(35,50): The message in the quote field is followed by the codes of the three special characters 219,224, and 251 with space codes between them. The label mode is ended with "z".



Graph Command GR

The graph command enables HP-IB input to be plotted as a trace. That is, amplitude inputs in display units are input starting at the left of the display. For each y input, x is automatically incremented by one display unit.

Syntax

GR $y_1, y_2, y_3, y_4 \ldots$

The GR command instructs the analyzer to graph the next points input as amplitude coordinates in the trace C display memory. The first point, y_1, will be at the left of the display, and successive points will be plotted left to right in the graticule space (display size D1). Lines are drawn between successive points. Trace C is set to view.

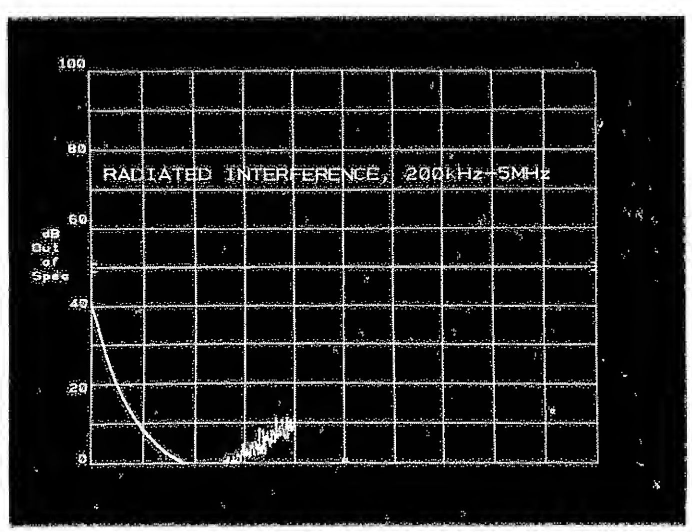
Example

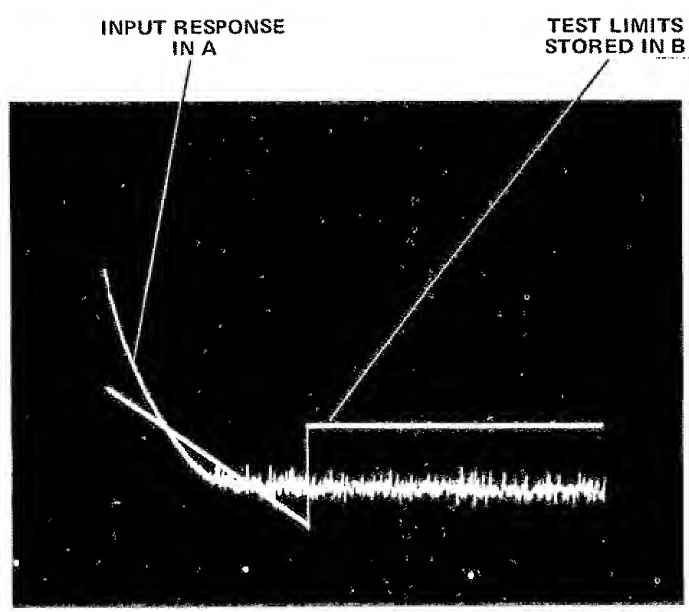
The test limits for an electromagnetic interference test are graphed into trace C using GR. The interference signal is input to trace A. Subsequent B ⇄ C exchange and A – B→A will plot only the out-of-spec interference signals on the CRT. Special annotations are added.

```
0: fmt
1: fmt ;wrt 718,"IP FA200KZ FB5MZ S2 GR"
2: for N=1 to 400
3: wrt 718,400-(3.5/4)N
4: next N
5: for N=401 to 1000
6: wrt 718,300
7: next N
8: wrt 718,"KSi TS KSk B3 C2 TS"
9: fmt -f4.0
10: wrt 718,"HD EM KSo DT?"
11: for N=1 to 11 by 2
12: wrt 718,"D2 PU PA 50",90N-20,"LB",10N-10,"?"
13: next N
14: wrt 718,"B4"
15: wtb 718,"PU PA0,600 LB   dB",10,13,"
    Out",10,13,"  of",10,13," Spec?"
16: wrt 718,"D3 PA100,500 LBRADIATED INTERFERENCE,
    200kHz-5MHz?"
```

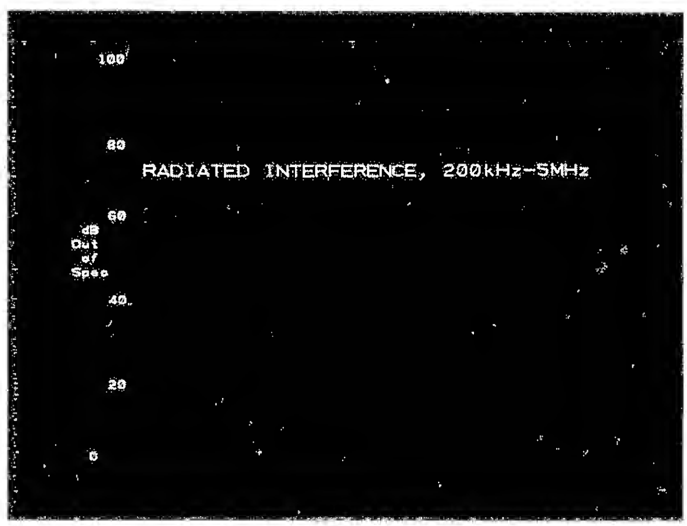
Line 1:	Initiates the graph mode. The IP set DA to 3072 so the graphing starts at the beginning of trace C.
Lines 2 to 7:	Writes the test limit values into the trace C memory.
Line 8:	Puts the graph data into trace B memory and enables A – B→A .
Line 9:	Format so that the CRT numbers will not have digits to the right of the decimal.
Line 10:	Clears the active function readout HD, prepares trace C for input EM, clears the display annotation KSo, and sets the label terminator to ?.
Lines 11 to 16:	Labels the graticule.
Line 15:	CR/LF (ASCII codes 13 and 10) are used to write on the next line. Note the binary write controller statement wtb.

The results of this test show the amount of radiation over the test limit versus frequency.





The original test limits and input can be recalled by the sequence
[LCL] [SHIFT] k [SHIFT] m
A – B [OFF]
[VIEW] A [VIEW] B



Trace C stores the special annotation. (This display can be obtained by [LCL] [BLANK] A [BLANK] B [BLANK] j.)

Input Trace B IB

The input trace B command allows input of all 1001 trace B points in two byte binary format 02. A trace is stored in controller memory using "02TB", the output of trace B in 2 byte binary format, then recalled with IB.

Example

The following program saves a trace in B[1001,2] controller memory array, then restores it to trace B using IB.

```
0:  fmt
1:  dim B[1001,2]
2:  wtb 718,"02TB"
3:  for N=1 to 1001
4:  rdb(718)→B[N,1]
5:  rdb(718)→B[N,2]
6:  next N
7:
8:
9:
10:                       measurement program
11:
12:
13:
14:  stp
15:  wtb 718,"IB"
16:  for N=1 to 1001
17:  wtb 718,B[N,1],B[N,2]
18:  next N
```

Lines 0 to 6:	Stores trace B (in binary) in B[ ].
Lines 15 to 18:	Restores trace B.
Line 15:	The command IB must not be followed by a CR/LF, thus the use of wtb.
Line 17:	Writes two binary numbers for each display point.

Note: Another command, KS 125_{10} can be used to store trace data in a similar manner at any display location. See page C.4.

Chapter 5

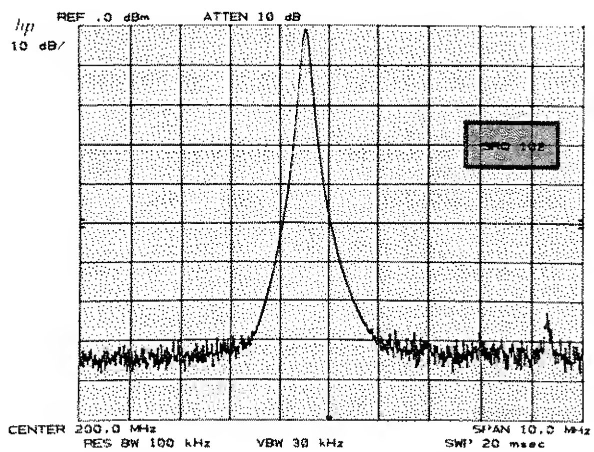
SERVICE REQUESTS

This chapter describes the analyzer's request for service capability and its use for interruptions to obtain service from an HP-IB controller.

A service request is an analyzer output which signifies an occurance at the analyzer, such as a units key pressed, the end of a sweep, or a hardware problem. A service request may trigger the controller to take action, such as changing the instrument state or writing data into the display memory. Service requests place the HP-IB SRQ line true.

When a request for service is being made, the CRT display reads out "SRQ" with a number.

NOTE
If the CRT display annotation has been blanked, the service request notation will **not** appear.



Display During A Service Request

Whether or not the SRQ message is displayed during a request for service, the HP-IB service request line (SRQ) is pulled true, announcing to the HP-IB controller that the analyzer requires attention. The analyzer sends a status byte on the bus which can be interpreted by the controller.

NOTE
HP-IB controller must use a serial polling technique to test for service requests. The analyzer will not respond to HP-IB parallel polling.

Status Byte Definition

Bit	Message	CRT Display Message
0 (LSB)	Unused.	—
1	Units key pressed.	"SRQ 102"
2	End of sweep.	"SRQ 104"
3	Hardware broken.	"SRQ 110"
4	Unused.	—
5	Illegal analyzer command.	"SRQ 140"
6	Universal HP-IB service request. HP-IB RQS Bit.	—
7	Unused.	—

The CRT SRQ number is an octal number based upon the status byte's binary value: For example, the status for an illegal analyzer command is

bit number	7	6	5	4	3	2	1	0
status byte	0	1	1	0	0	0	0	0

What appears on the CRT display is the octal equivalent of the status byte's binary number:*

"SRQ 140"

The octal number will always begin with a "1" since this is translated from bit 6, the universal HP-IB service request bit.

The decimal equivalent is $1 \times (8)^2 + 4 \times (8)^1 + 0 \times (8)^0 = 96$.

More than one service request can be output at the same time. For example, if an illegal analyzer command and an end-of-sweep occurred at the same time "SRQ 144" would appear.

bit number	7	6	5	4	3	2	1	0
status byte	0	1	1	0	0	1	0	0

"SRQ 144"

* The octal equivalent is based upon whole number:

001100000 (binary); $= 1 \times 2^5 + 1 \times 2^6 = 96$ (decimal)

However, one simple way to determine its octal equivalent is to partition the binary number 3 bits at a time from the least significant, and treat each section as a binary number alone. Thus

binary	01	100	000
octal	1	4	0

Service Request Commands R1, R2, R3 and R4

Except for the illegal command service request, SRQ 140, requests for service will not occur unless the appropriate activating command has been given.

SR Command	Allows	Cancelled By
R1	SRQ 140 only (illegal command)	none
R2	SRQ 140 SRQ 104 (end of sweep)	R1 only
R3	SRQ 140 SRQ 110 (hardware broken)	R1 only
R4	SRQ 140 SRQ 102 (units key pressed)	R1 or pressing units key

Commands R2, R3 and R4 may be activated simultaneously, allowing all the SRQ's. R4 must be re-enabled after its use or whenever any SRQ is cleared. R2 and R3 remain enabled until disabled by R1; in other words, R1 is used to disable service request commands except SRQ 140.

An instrument preset enables R3.

R2 also causes an SRQ at the end of the calibration routine and when the specified number of video averages is reached.

Controller Interrupt with Service Request

The HP-IB controller response to a service request depends upon the controller. The operating manual for each controller's HP-IB interface discusses reaction to a pulled SRQ line.

Computing controllers, such as the HP 9825A and HP 9830A, have commands which allow monitoring the SRQ line, then interpreting and clearing the status byte if a request for service occurs.

9825A Computing Controller Statement Review

Bit Functions

bit	bit (N,A)	Returns the value of the Nth bit in A (0 or 1).
	bit ("1Ø1ØXX",A)	Returns 1 if the mask matches the bit pattern in A, or 0 if the mask does not match. X or other character in the mask indicates bit which is not checked.
dto	dto A	Returns the octal equivalent of the decimal value specified by A.

Interrupt Statements

oni	oni 7, "shutoff"	Establishes the service routine where program execution will branch to interrupt on SRQ from the HP-IB specified by select code 7.
eir	eir 7	Enables the calculator to accept an SRQ interrupt from the HP-IB specified by select code 7.
	eir 7,M	
iret	iret	Signals the end of a service routine. During the interrupt service routine, the interrupt for the peripheral being serviced is automatically disabled to prevent cascading of interrupts.

Examples

This program includes an SRQ interpreter which prints the name of the service request enabled.

```
0: fmt
1: wrt 718,"R1 R3 R4"
2: oni 7,"Interpret SRQ"
3: eir 7
4:
5:
6:
7:                                    any program
8:
9:
10:
11: end
12: "Interpret SRQ":
13: rds(718)→S
14: if bit(3,S);prt "hardware broken"
15: if bit(5,S);prt "illegal command"
16: if bit(1,S);prt "units key pressed"
17: wrt 718,"R4"
18: eir 7
19: iret
```

Line 1:	All but the end of sweep SRQ's are enabled. R1 cleans out former SRQ commands
Line 2:	The "Interpret SRQ" subroutine will be executed when an SRQ occurs.
Line 3:	The controller's interrupt capability is enabled and the program continues.
Lines 12-19:	Interrupt subroutine.
Line 13:	The status byte is read into S and the SRQ line is cleared.
Lines 14-16:	The octal status byte is compared to each analyzer SRQ code. If true the name is printed on the printer.
Line 17:	R4, the units key SRQ is re-enabled.
Line 18-19:	The interrupt capability is re-enabled and the mainline program is continued.

SERVICE REQUESTS

In the following program, data is recorded only when the first sweep ends. This ensures that the test data is complete.

```
0: fmt
1: "Main program":
2: wrt 718,"R2"
3: oni 7,"record data"
4: eir 7
5:
6:                         } any program
7:
8:
9: end
10: "record data":
11: rds(718)→S
12:
13:
14:                        } data output subroutine
15:
16:
17: wrt 718,"R1"
18: iret
```

Line 2:	The end of sweep SRQ is enabled.
Lines 10-18:	Record data subroutine called when the sweep ends.
Line 11:	Reads the status byte and clears the SRQ line.
Line 17:	End of sweep SRQ is cleared by R1; and SRQ will not be called at the next end of sweep.

The same R2 command can be used to ensure the marker is placed before data is output in the program.

The following program uses R4 to allow a data entry into the controller from the analyzer DATA keyboard. Such an entry allows branching to other programs or changes to the instrument state.

```
0: fmt
1: "Main program":
2: oni 7,"key"
3: eir 7
4: wrt 718,"R1 R4"
5: wrt 718,"EE"
6:
7:
8:                         } any program not addressing the analyzer
9:
10:
11: end
12: "key":
13: rds(718)→S
14: wrt 718,"OA"
15: red 718,X
16: dsp X
17:                        } any useful subroutine using variable X.
18:
19: wrt 718,"R4EE"
20: eir 7
21: iret
```

Line 4: Disables previous SRQ commands with R1, then enables the units-key-pressed SRQ command, R4.

Line 5: Enables entry from the DATA keyboard. When an entry is completed by pressing the units key, "key" subroutine is called.

Lines 6-10: Computation or control not involving the analyzer.

Lines 12-21: The SRQ subroutine clears the SRQ line, reads the data entered (OA) and enters the value into variable X.

Service Request From The Front Panel

The operator can call for service from a controller from the front panel when in local by pressing

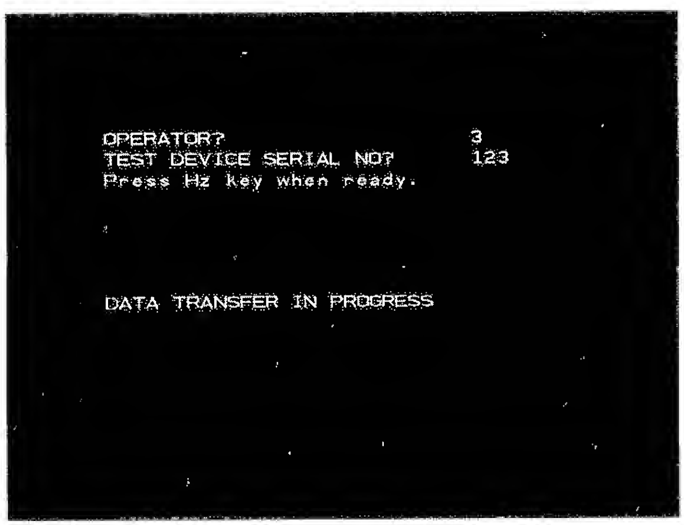
SHIFT r

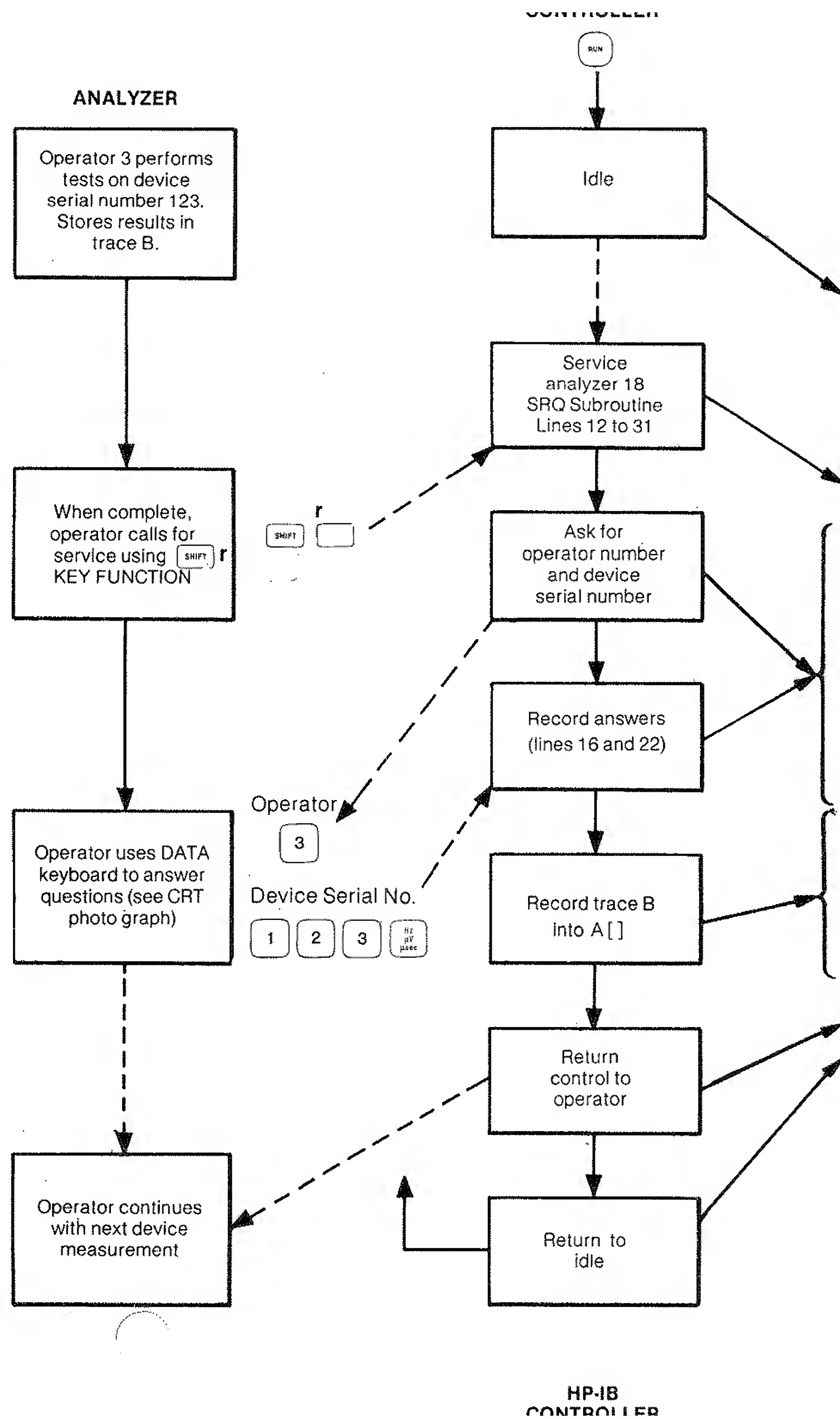
This front panel request for service sends SRQ 102, the units-key-pressed SRQ. It is not necessary for the SRQ command R4 to be enabled to use the front panel service request.

Example

One use for the front panel service request is to summon a remote controller for assistance. Several analyzers, each with a different HP-IB address, can call for a service such as recording trace test data. The following example suggests one possible way to do this for a single analyzer.

During the data transfer, beginning at line 24, the CRT display will appear as follows, with the "DATA TRANSFER" message blinking.






```
0: fmt
1: dim A[1001],A$[20]
2: "  idle  ":
3: oni 7,"data";eir 7
4: lcl 718
5: eir 7;rds(718)→S;if not bit(1,S);jmp 0
6: gto -1
7: red 718,A
8:
9: end
10:
11:
12: "data":fmt f.0
13: rds(718)→D
14: wrt 718,"SV1 EM O1 KSm KSo A4 DT:"
15: wrt 718,"D3 PUPA64,544 LBTEST STATION?:"
16: wrt 718,"EEOA";red 718,S;if S=0;jmp 0
17: wrt 718,"D3PUPA512,544LB",S,":"
18: wrt 718,"D3 PUPA64,512 LBTEST DEVICE SERIAL NO?:"
19: wrt 718,"D3 PUPA64,490 LBPress Hz key when ready.:"
20: wrt 718,"R1R4EE"
21: rds(718)→D;if bit(1,D)=0;jmp 0
22: wrt 718,"OA";red 718,A;str(A)→A$
23: wrt 718,"PUPA512,512LB",A$[2,len(A$)-3],":"
24: wtb 718,"PUPA 64,312LB",17,"DATA TRANSFER
                                          IN PROGRESS",18,3
25: wrt 718,"TB"
26: for N=1 to 1001
27: red 718,A[N]
28: next N
29: wrt 718,"EM RC1"
30: lcl 718
31: iret
```

other lines:
line 13 - reads and clears service request
line 29 - erases CRT questions

Appendix A

DISPLAY MEMORY STRUCTURE

This appendix discusses the details of the display memory as background for advanced HP-IB display programming (Appendix B). A summary of trace data manipulation by the trace mode functions is also discussed.

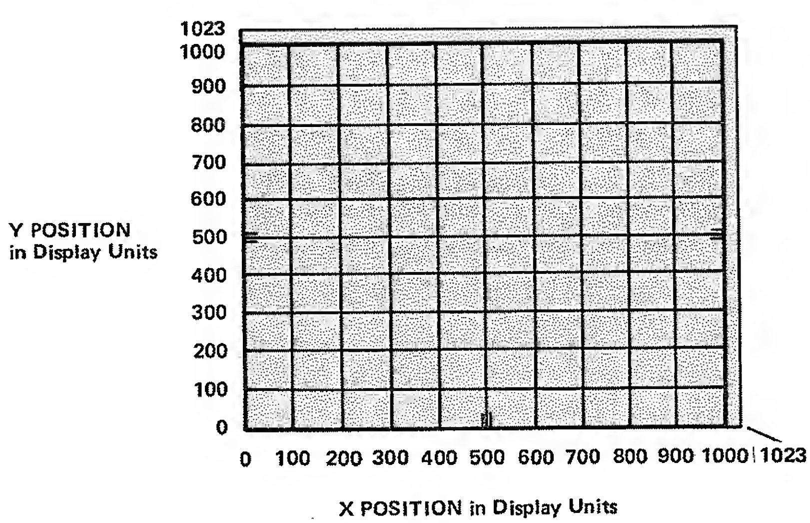
The display memory is defined as the digital storage allocated in the spectrum analyzer for the information which is presented on the CRT display. It is comprised of four different memories: three trace memories and one annotation memory. Addresses are assigned as follows:

DISPLAY MEMORY	ADDRESSES
Page 1 Trace A	Ø – 1023
Page 2 Trace B	1024 – 2047
Page 3 Graticule Annotation	2048 – 3071
Page 4 Trace C	3072 – 4095

Traces

The trace pages are used primarily to store analyzer response data to be displayed. Use is not restricted to the storage of trace data. As Chapter 4 describes, operator defined graphics and annotation can be written into the memory for display on the CRT.

Each trace address may contain an integer from 0 to 4095. When drawing trace values from 0 to 1023 will be plotted on the CRT display as amplitude y position, in display units. Appendix B discusses these values in detail.



For traces A, B and C the horizontal distance on the CRT is determined by the amplitude value's proximity to the first trace address, in the example below, in address 1024.

Trace B (Page 2) 1024 Addresses

Address	Amplitude Value, Y	(x,y) Position on CRT
1024	1040	Display Instruction
1025	622	(0,622)
1026	531	(1,531)
¦	¦	¦
2023	181	(998,181)
2024	162	(999,162)
2025	185	(1000,185)
2026	1072	Overrange Blanked
2027	1072	
¦		
2046	1072	
2047	1072	
3071		

The addresses 2023 and 2024 describe one trace line, drawn between (998,181) and (999,162). The values in the X-overrange addresses blank those lines.

Annotation and Graticule

Page 3 of the display memory is filled with instructions upon instrument preset. These instructions draw the graticule and annotation on the displays.

The display memory contents for the addresses in page 3 are listed in the table below. The first address given on each line is that of the instruction for the specific readout.

Address	Contents*
2048-2064	controls marker, display line, threshold annotation and graticule on/off functions
2065-2079	center line marks
2085-2099	marker symbols
2100-2114	display line
2115-2167	graticule
2168-2175	"hp"
2176-2192	"BATTERY"
2193-2208	"CORR'D"
2209-2240	"RES BW"
2241-2272	"VBW"
2273-2304	"SWP"
2305-2336	"ATTEN" and "PG" preamp gain number
2337-2368	"REF"
2369-2384	"dB/", "LINEAR"
2385-2400	trace detection mode: "SAMPLE", "POS PK", "NEG PK"
2401-2432	"START" or "CENTER"
2433-2464	"STOP" or "SPAN"
2465-2496	"OFFSET" for amplitude
2497-2528	"DL"
2529-2560	"TH"
2561-2624	"MKR" or "CNTR" or "MKR Δ"
2625-2656	"OFFSET" for frequency
2657-2668	"VID AVG"
2689-2751	title
2753-2768	"YTO UNLOCK"
2769-2784	"249 UNLOCK"
2785-2800	"275 UNLOCK"
2801-2816	"OVEN COLD"
2817-2832	"EXT REF"
2833-2848	"VTO UNCAL"
2849-2864	"YTO ERROR"
2865-2879	"MEAS UNCAL" or "*"
2880-2944	frequency diagnostics
2945-2960	"2ND LO", "↑","↓"
2961-2976	"SRQ" number
2977-3007	center frequency "STEP"
3008-3071	active function readout

*" " indicate the CRT annotation stored, values included where applicable.

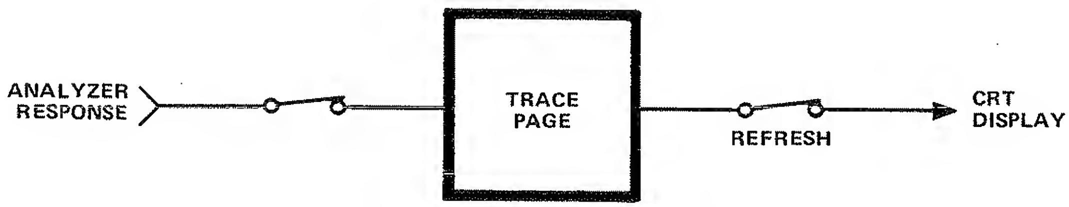
Data Transfer

The trace functions dictate the way in which data is input to and output from the trace page.

This section describes each TRACE function in terms of the interactions of the analyzer response, trace page and CRT display. The events are listed in chronological order, starting from when the trace function is activated. In each case, the analyzer accepts the function command immediately.

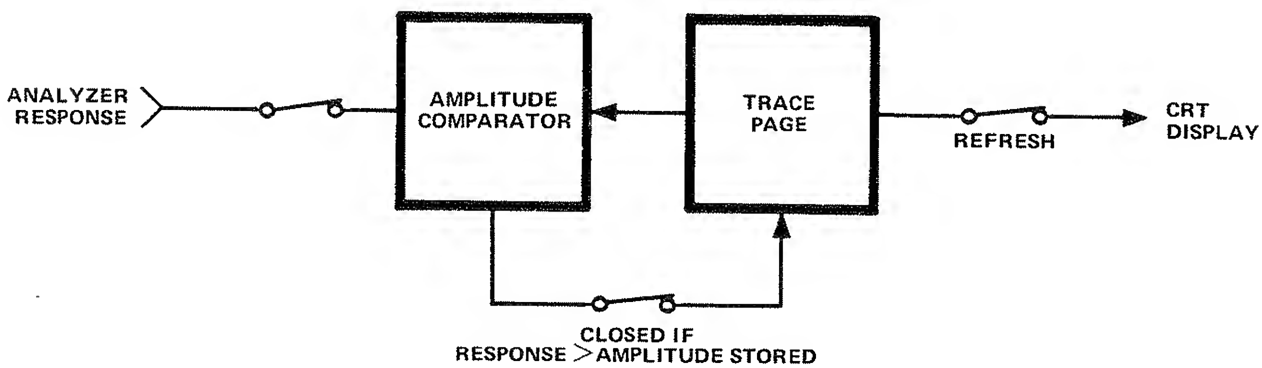
Clear-Write

1. Sweep is stopped.
2. Zero is written into each trace address and displayed on the CRT in one refresh.
3. On the next trigger, the sweep is started from the start frequency (CRT display left), and the amplitudes are written into memory.



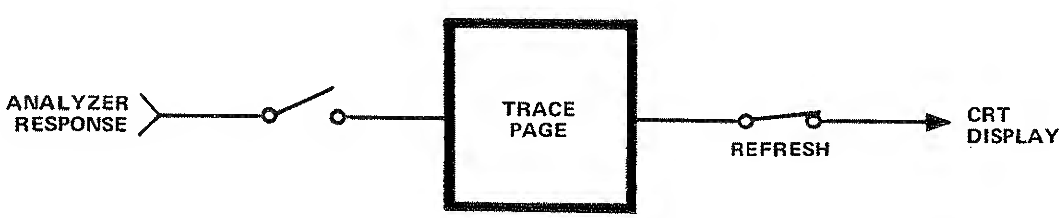
Max Hold

1. Sweep is stopped, but restarts from the left on the next trigger.
2. During each subsequent refresh, the amplitude stored at each trace memory address is compared to the corresponding analyzer response. The largest of the two will be stored at the trace address.



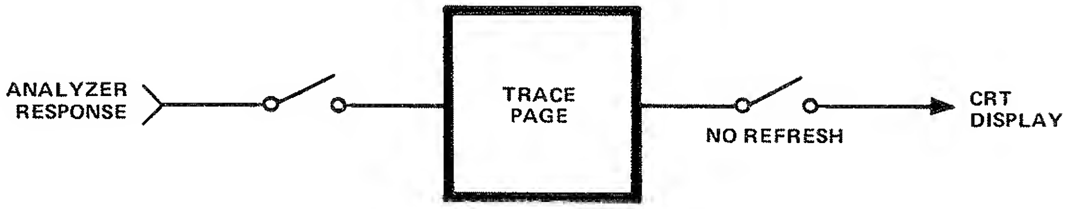
View

1. The sweep is stopped and the trace is displayed on the CRT.



Blank

1. The sweep is stopped and the trace is not displayed.



Exchange A and B

1. The sweep is stopped. If either trace is in a WRITE mode, it is placed in view.
2. The contents of traces A and B are exchanged.

A – B→A On

1. The sweep is stopped and trace B is placed in a STORE mode.
2. A is replaced with A-B.
3. The sweep is continued from where it stopped. Each new analyzer response point is reduced by the amount stored in the corresponding address of trace B, and the result is stored in trace A. This process continues at the sweep rate.
4. Subsequent sweeps will continue the process.

A – B→A Off

1. The analyzer response is written directly into trace A. Trace B and mode are not changed.

B – DL→B

1. Trace B is placed in view. Trace A is not changed.
2. The amplitude stored in the display line register is subtracted from the contents in each trace B address and the result stored at the same trace B address.

Appendix B**

ADVANCED DISPLAY PROGRAMMING

This appendix describes the explicit CRT display programming possible with the analyzer's display language.

A display program allows additional graphics capability on the CRT of the spectrum analyzer beyond those discussed in Chapter 4. Explicit display programming uses less display memory, allowing more efficient use of the 4096 display addresses available.

Appendix A, Display Memory Structure, provides background material for this section.

Display Program

A display program consists of a specific set of display instructions and data words written into the display memory. The **display instructions** dictate the operating mode of the CRT circuitry, such as label, graph or plot. The **data words** direct the CRT beam according to the preceding instruction.

Display instructions and data words are written into memory when the display programming codes are used. For example, the code "PA 500,600" writes into the display memory the instruction code for vector, 1026, followed by the x and y data values 500 and 600. This same "plot absolute" command could just as well been written as a display program by writing "1026, 500, 600" into the display memory. The display program is "executed" over and over to refresh the CRT from memory.

The commands necessary for writing display programs into memory are:

DA	<display address> puts the address into the display address register (referred to as the current address). (Display address means display memory address.)
DW	<instruction or data write> writes the instruction or data word into the current display address.
DD	<binary instruction or data write> writes the two 8 bit binary words into the current address.*
DR	<display read> outputs the contents of the current address on to the HP-IB data lines to be read by the HP-IB controller.

Appendix C discusses the syntax of each of these commands in more detail.

Loading and Reading a Display Program

Instructions and data words are loaded directly into the analyzer's display memory by, first, specifying the beginning address of the program, then writing in the instructions and data serially. To write the "1026, 500,600" program beginning at address 1024, which is the first address of trace B, execute

```
0: wrt 718,"DA1024DW1026;500,600" **
```

This display program instructs the display to draw a vector to the position (500,600) on the CRT.

To read and print out the display program, run:

*The first byte contains the four most significant bits, the second contains eight least significant bits of the 12 bit instruction or data word.

**All examples in Appendix B use the 9825A Computing Controller.

```
0: fmt
1: wrt 718,"O1DA1024"
2: for I=1 to 3
3: wrt 718,"DAOA"
4: red 718,A
5: wrt 718,"DR"
6: red 718,W
7: wrt 6,A,W
8: next I
9: end
```

Address	Contents
1024.00	1026.00
1025.00	500.00
1026.00	600.00

Line 1: Formats for decimal word values, sets the first address to be read.

Lines 2-8: Read and print three successive display program addresses and their contents. The address is automatically incremented by one for each DR execution.

Line 3: Calls for the output of the display address.

Line 5: Calls for the output of the current display address contents.

The following sections define and outline the instruction and data words. The final section provides detailed examples of display programming and a consolidated coding sheet.

Display Program Word

A display program word can be a value from 0 to 4095. The value is stored as a 12 bit binary word, and several of the bits define the type of word. Graphic representations used in this appendix are defined as follows:

	MSB											LSB	
bit	11	10	9	8	7	6	5	4	3	2	1	0	Decimal total:
	0	1	0	X	X	X	X	X	X	X	1	0	1026 +
decimal value	2048	1024	512	256	128	64	32	16	8	4	2	1	

where x is either a 1 (true) or a 0 (false).

The sample word displayed is 1024 + 2 = 1026, the instruction control word for vector used in the previous examples.

Instruction Words

There are three types of instruction words:

		11	10	9	8	7	6	5	4	3	2	1	0	
1:	**Display control**	0	1	0	X	X	X	X	X	X	0	X	X	1024 +
2:	**Program control**	0	1	0	X	X	X	X	X	X	0	1	1	1027 +
	including end of display	0	1	0	X	X	X	X	X	X	1	X	X	1028 +
3:	**Count/Threshold**	0	1	1	X	X	X	X	X	X	X	X	X	1536 +

Display Control Instructions

The display control instruction instructs the CRT circuitry on how to use the subsequent data words to direct the CRT beam. The word 1026, vector, is an example. Data values in a display program following 1026 will direct the CRT beam to x,y positions. The two other display control words are label, which writes characters on the CRT, and graph, which displays traces.

	11	10	9	8	7	6	5	4	3	2	1	0	
vector (vtr)*	0	1	0	X	X	X	0	X	X	0	1	0	1026 +
label (lbl)	0	1	0	X	X	X	0	X	X	0	0	1	1025 +
graph (gra)	0	1	0	X	X	X	0	X	X	0	0	0	1024 +

where + indicates that additional bits can be set without changing the primary function.

The syntax of vector, label and graph are similar to their programming code counterparts PA/PR, LB and GR, respectively. Pen up/down, changing display size and beam intensity are controlled by setting various bits along with the control instructions and data word. These functions are called ancillary functions to the instruction.

	11	10	9	8	7	6	5	4	3	2	1	0
ancillary functions	0	1	0	X	X	X	0	X	X	0	X	X

display size
big expand (**bex**), + 256
expand and shift (**exs**), + 64

beam intensity
dim (**dim**), + 8
bright (**brt**), + 128

clear x position (**clx**), + 16

*Abbreviations within the parenthesis are useful as a short hand notation for writing display programs. They are not programming codes.

clear x position (**clx**):	The x axis display position is reset to the far left (0,y).
big expand (**bex**):	The x and y CRT beam deflection is amplified by a 1.49 factor.[1]
expand and shift (**exs**):	The x and y CRT beam deflection is amplified by a 1.13 factor (expand) and the (zero, zero) reference point is shifted to the lower left of the CRT screen (shift).[1]
dim(**dim**):	Sets the CRT beam intensity below the normal level.[2]
bright (**brt**):	Sets the CRT beam intensity to the maximum level.[2]

Program Control Instructions

The display program will normally execute the contents of memory starting with address 0 and working consecutively to address 4095. Program control instructions are used to alter the normal flow of a program by allowing program execution to be transferred anywhere in memory. These program control words allow jumps to specific display addresses (jmp), jumps to a display program subroutine (jsb), returns (ret), skips to the next control instruction (skc) and a word that simulates a "for . . . next" loop, the decrement and skip on zero (dsz).

	11	10	9	8	7	6	5	4	3	2	1	0	
jump (jmp)	0	1	0	X	0	0	0	X	1	0	1	1	1035
jump to subroutine (jsb)	0	1	0	X	1	0	0	X	1	0	1	1	1163
return (ret)	0	1	0	X	1	1	0	X	1	0	1	1	1227
skip to next control instruction (skc)	0	1	0	X	X	X	0	X	0	0	1	1	1027
skip to next memory page (skp)	0	1	0	X	X	X	1	X	0	0	0	0	1056
decrement and skip on zero (dsz)	0	1	0	X	0	1	0	X	1	0	1	1	1099

The address to be jumped to is the contents of the memory word following the jmp or jsb instruction. For example, "1035, 2048" causes program execution to jump to address 2048. The address given should contain a control instruction, that is, an instruction whose three most significant bits are 0 1 0. (If the address does not contain a control instruction, the program will go to the first control instruction following the specified address.) A return (ret) causes the program execution to return to the first control instruction following the jsb instruction which sent it to the subroutine.

NOTE

Subroutines must not contain label or graph control words. A subroutine may not call another subroutine.

(1)The display size commands combine these size instructions as follows:

	instructions	ratio to D1	origin shifted
D1	none	1.00	no
D2	exs	1.13	yes
D3	bex and exs	1.68	yes
—	bex	1.49	no

(2)The intensity of the beam is also dependent upon line length. Lines longer than a preset length will be brighter because beam writing rate is slowed.

The skip to next control instruction (skc) causes program execution to go to the next control instruction in memory. The skip to next page (skp) instruction causes program execution to go to the next address which is an integer multiple of 1024. (An instruction which combines skp **and** skc, 1056 + 3 = 1059, will execute as if it were a skp followed by a skc.)

The decrement and skip on zero (dsz) instruction decrements an internal count register then tests the contents for zero. If the contents are not zero, the program goes to the next control instruction. If the contents equal zero, the program will skip the next two addresses then go to the next control instruction. For example, "1099, 1035, 1532, 1026" causes the program to skip to the control word 1026 if the counter register is zero; otherwise it executes the 1035, 1532 which is a jump to address 1532. See Count Register below.

The ancillary control function clear x position (clx) can be added to any of the program control instructions.

Another method of causing skips in program execution is in conjunction with the label mode (either LB or lbl). These are discussed in the Data Word section following.

End of Display

	11	10	9	8	7	6	5	4	3	2	1	0	
end of display (end)	Ø	1	Ø	X	X	X	X	X	X	1	X	X	1028+

When executed, the end of display instruction terminates execution of the display program. The next execution of the program will begin at display address zero upon the next display refresh trigger (note refersh trigger and sweep trigger are not the same).

The end of display instruction bit supercedes all other coding in the instruction except the ancillary function clear x position, clx (bit 4), which may be added. The end instruction will cause a default to graph mode at the beginning of the next program execution if no display control instruction is located at address zero.

Since the fast sweep (direct display of video and sweep) is displayed between program executions, an end instruction is required for the proper operation of the fast sweep display.

An end of display in trace C will be changed to a skip to next memory, 1056, when BC exchange is executed.

Count Register/Threshold

The load counter instruction loads an internal count register with a value determined by bits 0 through 8 of the instruction. The internal register is used in two ways. When in the graph (gra) mode, the display program interprets the register contents as the display THRESHOLD position. The second use is the count register for the decrement and skip on zero (dsz) instruction. The interpretation for these two uses is shown below:

	11	10	9	8	7	6	5	4	3	2	1	0	
load count register (ldc)	Ø	1	1	Ø	X	X	X	X	X	X	X	X	1536 + COUNT

COUNT (bits 7–0)
(1 to 255)

	11	10	9	8	7	6	5	4	3	2	1	0	
load threshold	Ø	1	1	Ø	X	X	X	X	X	X	X	X	1536+

Display threshold position
divided by 4.

	11	10	9	8	7	6	5	4	3	2	1	0	
threshold off	Ø	1	1	1	X	X	X	X	X	X	X	X	1792+

NOTE

The ldc and dsz instructions use the THRESHOLD level register. Therefore a load THRESHOLD instruction must be executed after all uses of ldc and dsz **and** before the next graph command is executed. Otherwise the threshold may not function correctly.

Data Words

Data words are differentiated by the two most significant bits. The following words are data words:

11	10	9	8	7	6	5	4	3	2	1	0	
0	0	X	X	X	X	X	X	X	X	X	X	0 to 1023
1	0	X	X	X	X	X	X	X	X	X	X	2048 to 3071
1	1	X	X	X	X	X	X	X	X	X	X	3072 to 4095

The use of these data word formats depend entirely upon the type of instruction word preceding.

Graph

Each data word following a graph instruction is interpreted as a y position. Y position values follow the general rule as shown below:

	11	10	9	8	7	6	5	4	3	2	1	0	
positive data, displayed	0	0	X	X	X	X	X	X	X	X	X	X	0 to 1023 = y position
positive data, blanked	1	0	X	X	X	X	X	X	X	X	X	X	2048 + y position
negative data, blanked	1	1	X	X	X	X	X	X	X	X	X	X	4096 - y magnitude (a two's complement value)

With negative data, the CRT beam goes to y = 0. Note that negative data can result from the trace arithmetic functions A – B→A and B – DL→B.

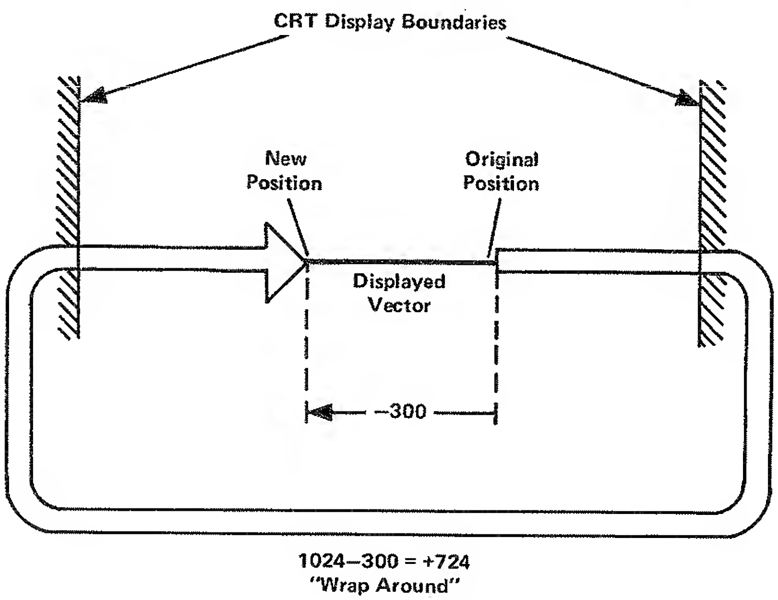
Vector

Data words following a vector (vtr) instruction are interpreted as x, y pairs. The data value determines whether the vector is blanked or displayed, absolute or relative. The x position data sets the absolute/relative ancillary function and the y position data sets the blank/unblank ancillary function.

	11	10	9	8	7	6	5	4	3	2	1	0
x position	R	0	X	X	X	X	X	X	X	X	X	X
y position	B	0	X	X	X	X	X	X	X	X	X	X

when	R = 1	(x position + 2048) vector is relative (both x and y are relative)
	R = 0	(x position + 0) vector is absolute (both x and y are absolute)
	B = 1	(y position + 2048) vector is blanked (pen up)
	B = 0	(y position + 0) vector is displayed (pen down)

Negative values for the plot relative x and y positions are entered as complementary values of 1024 to the ten least significant bits of the data word. For example, a plot relative −300 of x position is written in the data word as (1024 − 300) = 724. The actual plot "wraps around" the display to find the −300 position.



		10	9	8	7	6	5	4	3	2	1	0	
x & y negative relative	X	0	X	X	X	X	X	X	X	X	X	X	0-1023

1024 − |negative value|

Label

Data words following the label instruction are interpreted as character codes.

	11	10	9	8	7	6	5	4	3	2	1	0
character data word	X	X	X	X								

8 bit character code

The four most significant bits are not used. However, the two most significant bits must not be 0,1 respectively or the word would become a control instruction.

A specific set of character codes provide special label functions:

	Code
null	0
back space (BS)	8
line feed (LF)	10
vertical tab (opposite of line feed) (VT)	11
form feed (move beam to (0,0)) (FMFD)	12
carriage return (CR)	13
blink on (bkon)	17
blink off (bkof)	18
space (SP)	32
skip to next 16 block (sk16)	145
skip to next 32 block (sk32)	146
skip to next 64 block (sk64)	147

A blink on (bkon) will cause blinking of everything drawn on the display until a subsequent blink off (bkof) or an end of display (end) instruction is encountered with program execution.

A skip 16, 32 or 64 will cause program execution to go to the next address which is an integer multiple of 16, 32, or 64 respectively.

Note that these functions will work for both the lbl instruction code (1025 +) or the LB command.

Display Control Instruction Examples

These examples illustrate the use of display control instructions and data words. The loading and reading techniques described at the start of this appendix are used.

Graph (gra)

The graph instruction is used in the trace modes to plot the spectral traces on the CRT display. The graph display instruction along with ancillary functions can be used to visually modify stored trace data.

For example, in a trace stored in trace B memory, the portion between the 5th and 7th graticule line can be highlighted by making the trace brighter. First calculate the addresses where that trace data is stored

5th graticule address = 5(100) + 1024 = 1524
7th graticule address = 7(100) + 1024 = 1724
where 1024 is the first address of trace B.

Next, make up the graph instruction that will brighten the trace from the ancillary function codes:

1024 (gra) + 128 (brt) = 1152
1024 (gra) + 8 (dim) = 1032

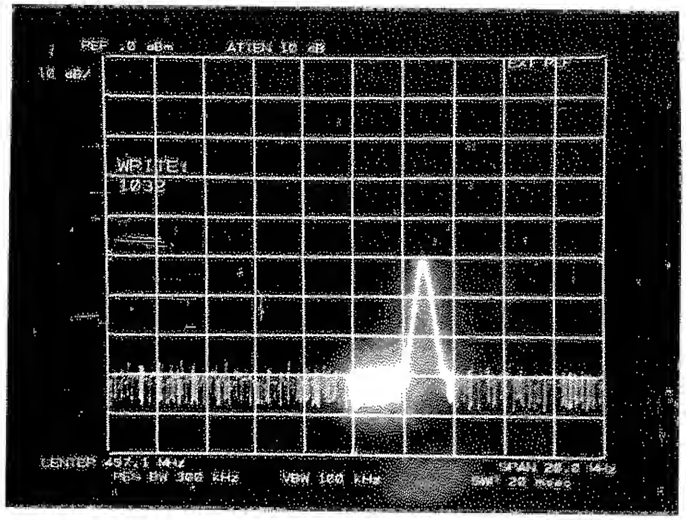
With trace B in view the loading program is

```
0: fmt
1: wrt 718,"DA1524DW1152"
2: wrt 718,"DA1724DW1032"
```

Line 1: Writes a graph/bright instruction so that every data point thereafter is brightened.
Line 2: Returns the beam intensity to dim.

Data points at 1524 and 1724 are lost and all data beyond 1524 is shifted 1 point to the left. All data beyond 1724 is shifted 2 points to the left.

These instructions will be written over when new trace information is written into trace B.



Vector (vtr)

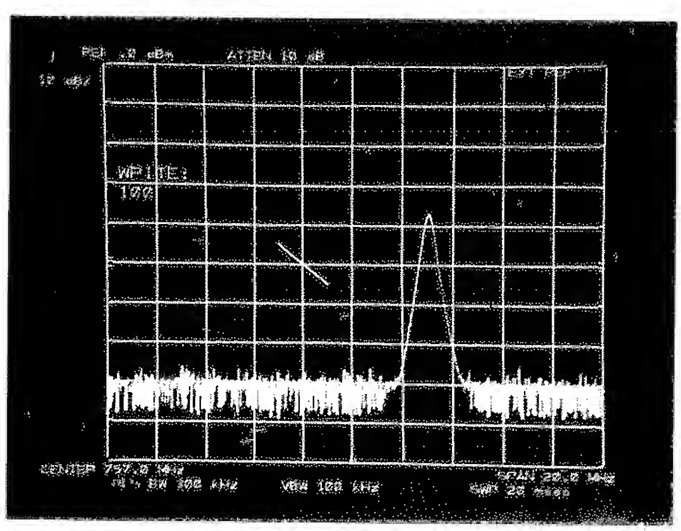
Instructions can be used to draw lines on the CRT display. The data words each determine whether the data is plotted absolute/relative or blanked/unblanked (pen up/pen down). The ancillary functions apply to the vector instructions.

For example, a line is to be plotted on the display with plot relative instructions in trace C memory beginning at address 3072.

address	description	program	word
3072	vector	vtr	1026
3073	x = 450 absolute	450 + 0	450
3074	y = 450 blanked	450 + 2048	2498
3075	x = −100 relative	(1024-100) + 2048	2972
3076	y = +100 relative pen down	100 + 0	100

The load program is:

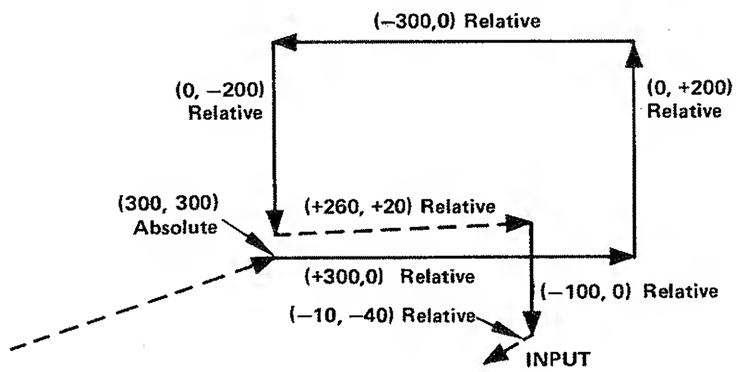
```
0: wrt 718,"DA3072DW1026,450,2498,2972,100"
```



Vector and Label (vtr and lbl)

To demonstrate the display instructions, a simple block diagram will be drawn and labelled. Then the control words will be modified with some of the ancillary functions to demonstrate their use.

First a graphics plan is drawn:



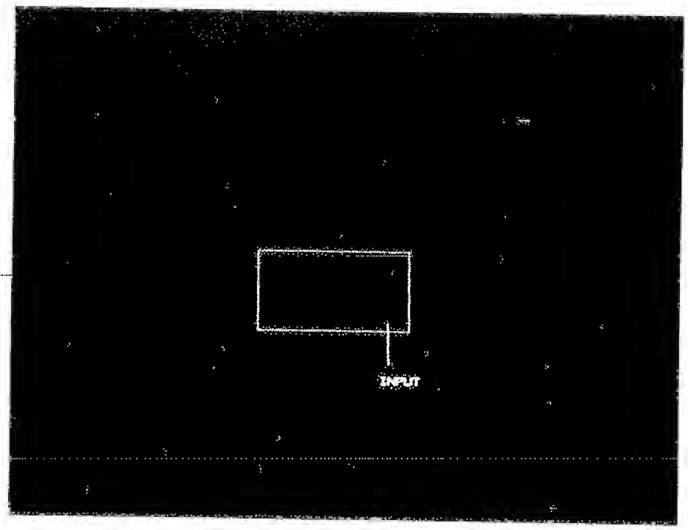
Graphics Plan

The vectors with + and − signs are relative vectors, the others are absolute points. Dashed lines are to be blanked.

address	description	program	word
3072	vector absolute	vtr	1026
3073	x = 300 absolute	300 + 0	300
3074	y = 300 pen up	300 + 2048	2348
3075	x = +300 relative	300 + 2048	2348
3076	y = 0 pen down	0 + 0	0
3077	x = 0 relative	0 + 2048	2048
3078	y = +200 pen down	200 + 0	200
3079	x = −300 relative	(1024-300) + 2048	2772
3080	Y = 0 pen down	0	0
3081	x = 0 relative	0 + 2048	2048
3082	y = −200 pen down	(1024-200) + 0	824
3083	x = +260 relative	260 + 2048	2308
3084	y = +20 pen up	20 + 2048	2068
3085	x = 0 relative	0 + 2048	2048
3086	y = −100 pen down	(1024-100) + 0	924
3087	x = −10 relative	(1024-10) + 2048	3062
3088	y = −40 pen up	(1024-40) + 2048	3032
3089	label	lbl	1025
3090	the word "INPUT"	I	73
3091		N	78
3092		P	80
3093		U	85
3094		T	84
3095	end of display	end	1028

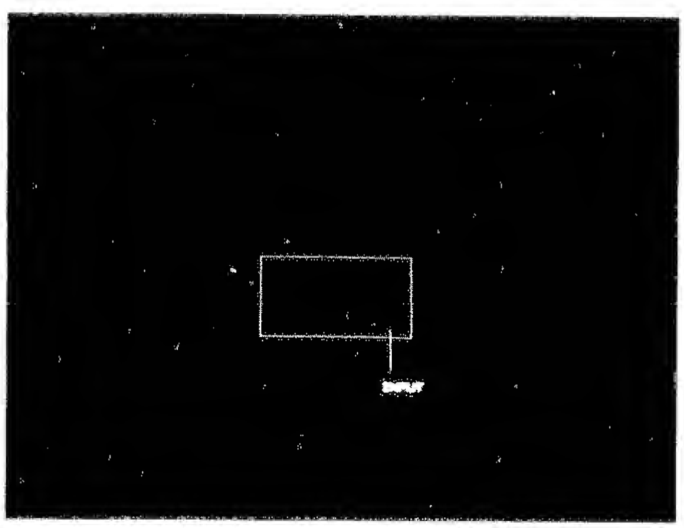
The above plan can then be programmed and run.

```
0: fmt
1: wrt 718,"IPKSoKSmA4"
2: wrt 718,"DA3072DW1026,300,2348,"
3: wrt 718,"2348,0,2048,200,"
4: wrt 718,"2772,0,2048,824,"
5: wrt 718,"2308,2068,2048,924,"
6: wrt 718,"3062,3032,"
7: wrt 718,"1025,73,78,80,85,84,1028,"
```



The display can now be modified by adding various ancillary functions to the existing control words. Brighten the "INPUT" term by adding 128 (brt) to the label address 3089 (1025 + 128 = 1153).

7: wtb 718, "LB",17,"INPUT",18,"1028,"



The label "INPUT" can be made to blink by adding blink on (bk on) and blink off (bk of) words before and after the "INPUT" label.

7: wrt 718, "1025,17,73,78,80,85,84,18,1028"

Alternately line 7 could have been written:

7: wtb 718, "LB", 17, "INPUT", 18, 1028,

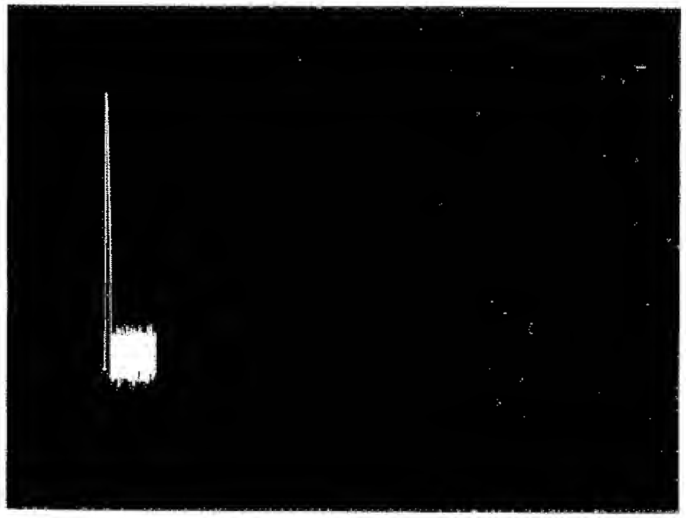
Program Control Instruction Examples

These examples use both program and display control instructions.

End of Display (end) and Skip to Next Memory Page (skp)

To end the display after the first 100 points of trace A write "DW 1028" into address 100.

wrt 718, "S2TSDA100DW1028"



All display memory information beyond the address 100 is ignored, including the annotation. Note that the analyzer sweep has been stopped with S2 to prevent signal response data from writing over the control word.

Skip control words allow certain portions of the display to be omitted from the display. There are two types of skip control words which enable 1) skip over the remainder of the present memory page to the beginning of the next memory, 2) skip to the next control word.

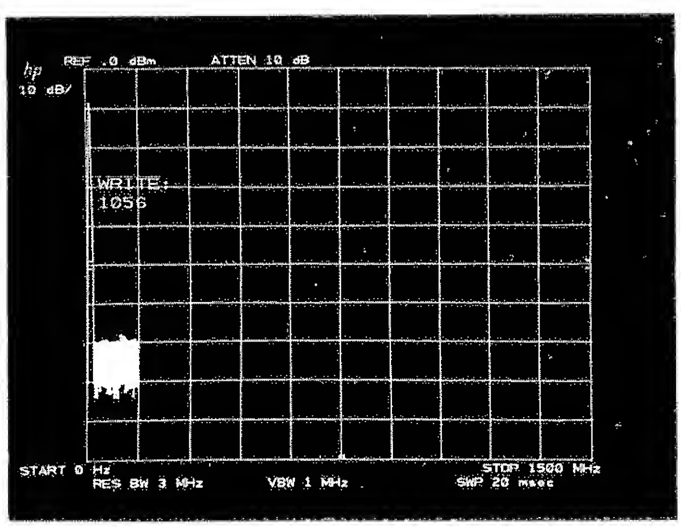
The skip page and skip to next control word have been assigned the two command codes PS and SW, respectively.

In the above example, the annotation was ignored because of the end of display written into address 100. If instead, a skp is written, the rest of the display memory can be displayed while omitting the remainder of trace A.

wrt 718, "S2TSDA100DW1056"

or

wrt 718, "S2TSDA100PS"*



*The programming code PS can be substituted for DW1056.

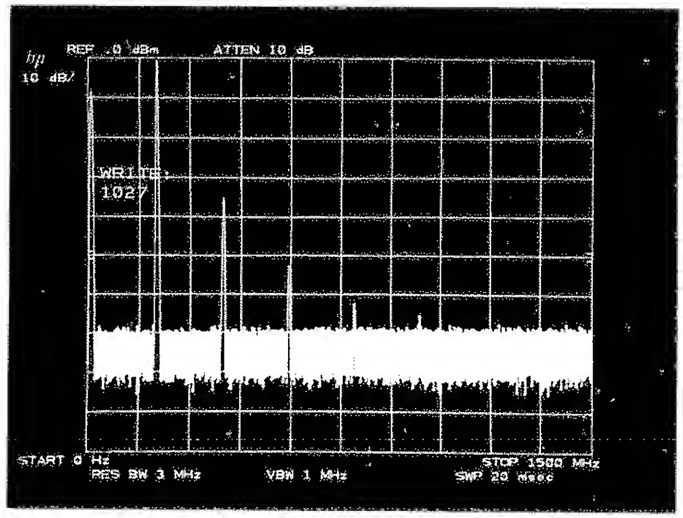
A skp written into the trace C page will skip the refresh pointer to DA 0 (trace A). This may cause an increase in the trace intensity since the program will not wait for a refresh trigger before beginning the next execution of the program.

Skip to Next Control Instruction (skc)

Program control is transferred to the next control instruction

For example, address 2073 of the annotation memory page contains the label control word which places the center frequency "| |" mark on the CRT. To omit this marker from the display, the label word is replaced by a skc word.

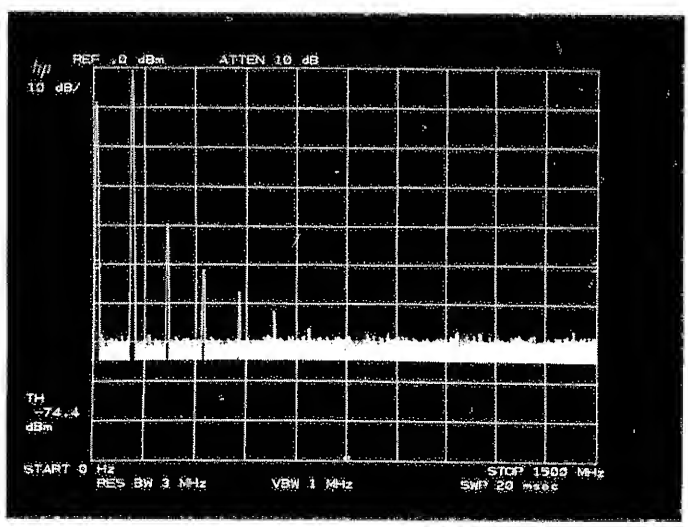
wrt 718, "DA2073DW1027"
or
wrt 718, "DA2073SW"*



Jump (jmp)

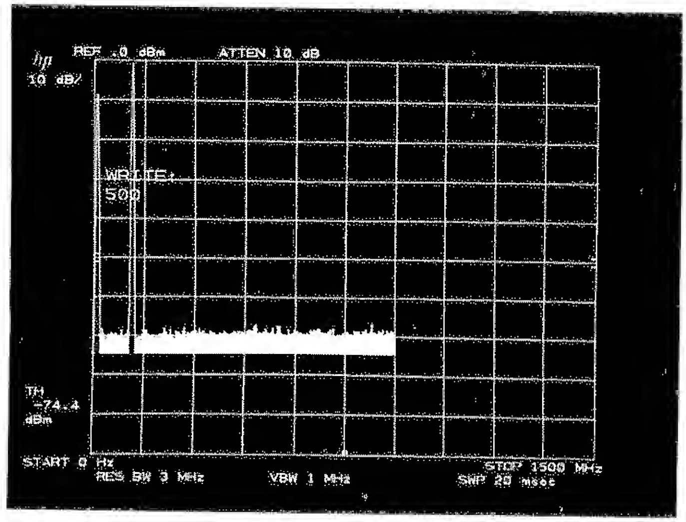
The example demonstrates jmp by jumping over the data in addresses 100 to 500 in trace A. Since the jump should be made to a control word, gra is first written into DA 500.

Before program is loaded.



*The programming code SW can be used for DW 1027.

wrt 718, "S2TSDA500DW1024"
wrt 718, "DA100DW1035,500"

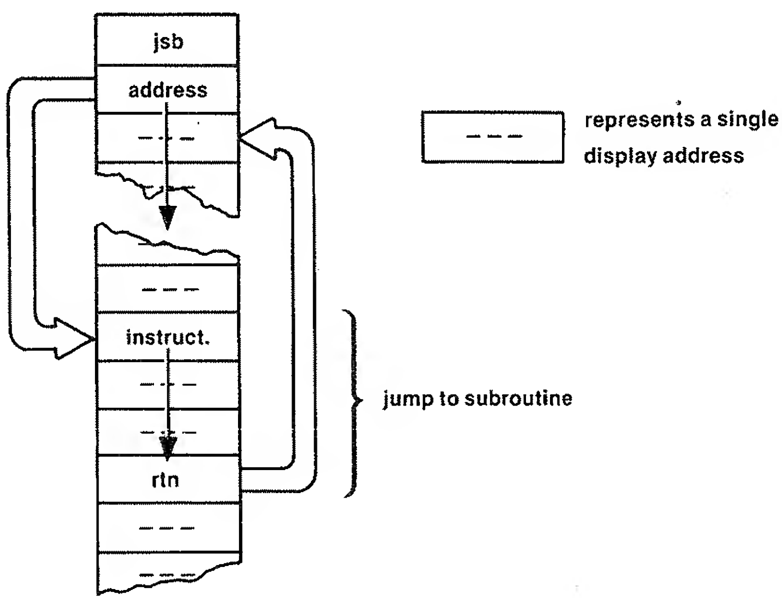


The signal response that would have been shown between display addresses 100 and 500 is omitted and the gap closed.

Jump Subroutine (jsb) and Return (rtn)

The jsb instruction transfers program control to the address specified. If the address does not contain a control word then the program will skip to the next control word after that address. The rtn instruction transfers program control to the first control word following the jsb instruction.

The flow of the program is as follows:



To demonstrate jsb/rtn, this example substitutes a new symbol for the preprogrammed marker symbol.

The marker symbol (a small diamond) is written as a subroutine in the annotation memory at address 2085. Substitution of the diamond symbol can be made by calling for and writing a new jsb routine with this program. The address for the marker subroutine call is located at display address 2054.

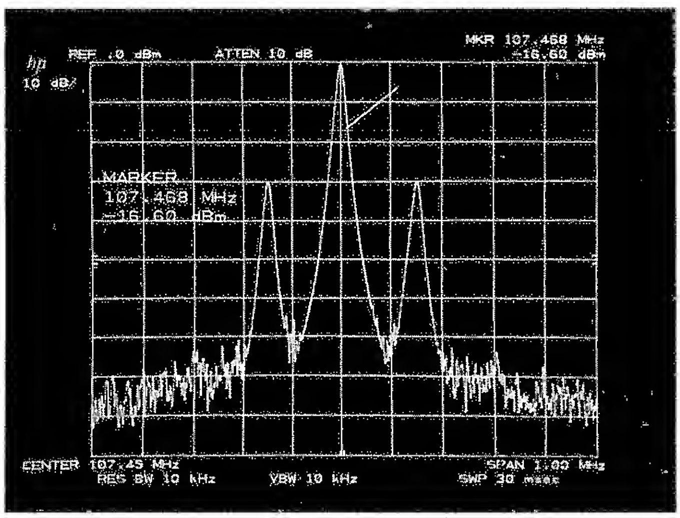
```
0: fmt
1: wrt 718,"DA2054 DW3080"
2: wrt 718,"DA3080 DW1154,2148,100"
3: wrt 718,"1227"
```

Line 1: Rewrites a new subroutine address, 3080, in place of the old one.

Line 2: Writes the new symbol vector subroutine starting at address 3080.

Line 3: Return

After running this program, the display memory contains the following:



Loop Instructions:
Load Counter Register (ldc) and Decrement and Skip on Zero (dsz)

As an example, looping will be used to draw a grid in two places on the CRT display in one refresh. The trace C page will be programmed to contain the graphics.

	address	description	program	word
positioning vector	3072	plot absolute	vtr	1026
	3073	x = 600 (PA)	600	600
	3074	y = 300 (PU)	300 + 2048	2348
	3075	jump sub	jsb	1163
	3076	to subroutine	address	3199
	3077	plot absolute	vtr	1026
	3078	x = 100 (PA)	100	100
	3079	y = 300 (PU)	300 + 2048	2348
	3080	jump sub	jsb	1163
	3081	to	address	3199
	3082	end of display	end	1028
looping subroutine	3199	vector	vtr	1026
	3200	repeat 10 times	ldc + 10	1546
	3201	plot relative	vtr	1026
	3202	x = 0 (PR)	0 + 2048	2048
	3203	y = +25 (PU)	25 + 2048	2073
	3204	x = +300 (PR)	300 + 2048	2348
	3205	y = 0 (PD)	0	0
	3206	x = 0 (PR)	0 + 2048	2048
	3207	y = +25 (PU)	25 + 2048	2073
	3208	x = −300 (PR)	1024-300 + 2048	2772
	3209	y = 0 (PD)	0	0
	3210	decrement	dsz	1099
	3211	jump to	jmp	1035
	3212	start	address	3201
	3213	return	rtn	1227

The program can then be written loading the words sequentially as listed in the above plan.

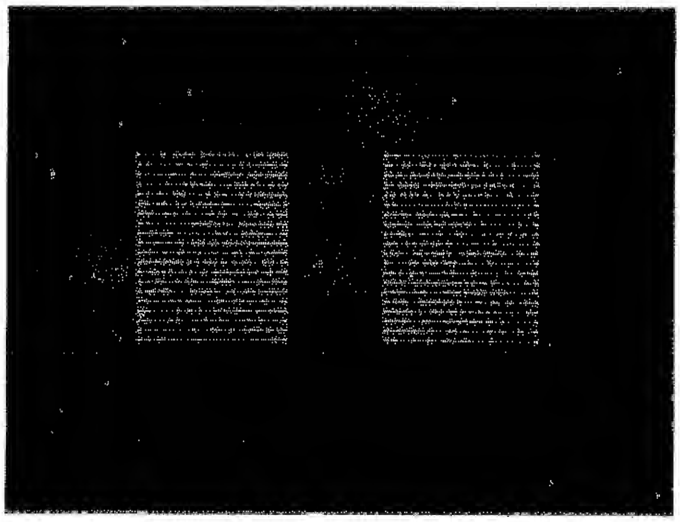
```
0: fmt
1: wrt 718,"IPKSoKSmA4"
2: wrt 718,"DA3072DW1026,600,2348,"
3: wrt 718,"1163,3199,1026,100,2348,1163,3199,"
4: wrt 718,"1028,"
5: wrt 718,"DA3199DW1026,1546,1026,"
6: wrt 718,"2048,2073,2348,0,2048,2073,2772,0,"
7: wrt 718,"1099,"
8: wrt 718,"1035,3201,"
9: wrt 718,"1227 HD"
10: end
```

Line 1: Initializes the analyzer.

Line 2 & 3: The positioning vectors.

Line 4: A skip to next memory insures that the following loop (DA 3199) is not refreshed unless called from addresses 3075 and 3080, the jsb words.

Running the program results in the following display:



INSTRUCTION AND DATA WORD SUMMARY

Display Control Instruction	Data	Word
graph (gra)		1024
	amplitude: position unblanked	y
	position blanked	y + 2048
	negative blanked	4096-|y|
label (lbl)		1025
	character	ASCII or special character code (≤255)
	blink on (bkon)*	17
	blink off (bkof)*	18
	skip to next 16 block (sk16)*	145
	skip to next 32 block (sk32)*	146
	skip to next 64 block (sk64)*	147
vector (vtr)		1026
	x position	data in display units
	y position	data in display units
	absolute vectors	x + 0
	relative vectors	x + 2048
	pen down	y + 0
	pen up (blanked)	y + 2048
Ancillary to gra, lbl and vtr instruction word:		
big expand (bex)		word + 256
expand and shift (exs)		word + 64
bright (brt)		word + 128
dim (dim)		word + 8
clear x position		word + 16
Program Control Instruction	**Data**	**Word**
end of display (end)		1028
skip to next memory page (skp)		1056 or "PS"
skip to next control word(1) (skc)		1027 or "SW"
jump(1)(jmp)		1035
	address	0 to 4096
jump to subroutine(1) (3) (jsb)		1163
	address	0 to 4096
return(1) (3) (ret)		1227
decrement and skip two addresses on zero(1) (2) (dsz)		1099
load counter (THRESHOLD position)(2)(ldc)		1536 + (count)

* These can also be accessed using the LB command. These functions can be initiated any time the label mode is active.

(1) Jumps and skips will skip to an address containing a control word.

(2) Loop should use only lbl and vtr control words. ldc is not a control word.

(3) Subroutines may use only vtr control words.

CONSOLIDATED CODING

INSTRUCTIONS

Display Control

bit number	msb 11	9	10	8	7	6	5	4	3	2	1	0
	0	1	0	X	X	X	X	X	X	X	X	X

- end of display (end) → bit 2
- dim (dim) → bit 3
- clear x position (clx) → bit 4
- skip to next page (skp) → bit 5
- expand and shift (exs) → bit 6
- bright (brt) → bit 7
- big expand (bex) → bit 8

bit 1	bit 0	
0	0	= graph (gra)
0	1	= label (lbl)
1	0	= vector (vtr)
1	1	= program control

Program Control

	bit 7	bit 6
jump (jmp)	0	0
decrement and skip on zero (dsz)	0	1
jump to subroutine (jsb)	1	0
return (rtn)	1	1

- bit 2: end of display (end)
- bit 3: 1 = jmp,dsz,jsb or rtn; 0 = skip to next control instruction (skc)
- bit 4: clear x position (clx)
- bit 5: skip to next page (skp)

Load Counter (ldc)

0	1	1	A	X	X	X	X	X	X	X	X

count (bits 7–0) — A = 0

Threshold

threshold position divided by 4 — A = 0
threshold off — A = 1

DATA:

Graph (gra)

		X	X	X	X	X	X	X	X	X	X

value (bits 9–0)

positive	0	0
positive blanked	1	0
negative blanked	1	1

Character

X	X	X	X								

where 0 1 not allowed; character code (bits 7–0)

Vector (vtr)

x position	R	0	X	X	X	X	X	X	X	X	X	X
y position	B	0	X	X	X	X	X	X	X	X	X	X

R = 1 relative vector
B = 1 blank vector

Appendix C

SYNTAX REQUIREMENTS

This appendix summarizes the syntax of the 8568A Spectrum Analyzer programming codes when controlled by an HP-IB Controller. HP-IB is Hewlett Packard's implementation of IEEE Standard 488-1975 and ANSI Standard MC1.1, "Digital Interface for Programmable Instrumentation".

Notation Conventions

< >	Items enclosed within angular brackets are considered to be elements of the language being defined.
: : =	"Is defined as": <A> : : = <B> <C> indicates that <A> can be replaced by the series of elements <B> <C> in any statement which <A> occurs.
[ ]	Square brackets indicate that whatever occurs within the brackets is optional.
|	"or": Indicates a choice of exactly one element from a list. (e.g., <A> | <B> indicates <A> or <B> but not both.

Definitions

<listen> : : =	Controller addresses analyzer to listen.
<talk> : : =	Controller addresses analyzer to talk.
<value> : : =	A number, either fixed or floating point format.
<address> : : =	The next display memory address to be accessed.
<format> : : =	01 | 02 | 03 | 04, the format of data output from the analyzer.

Data Entry to Analyzer

<entry> : : = <value> <terminator> | <units code> | <entry> <entry>

<terminator> : : = <units code> | <delimiter>

<delimiter> : : = <CR> | <LF> | , | ; | <ETX> (enters Hz, dB, volts or seconds)

<CR> : : = 13_{10} (ASCII carriage return)

<LF> : : = 10_{10} (ASCII line feed)

<ETX> : : = 3_{10}

<units code> : : = DM | – DM | DB | HZ | KZ | MZ | GZ | MV | UV | SC | MS | US

units of power:

DM : : = <dBm> | <dBmV> | <dBμV> | DB | GZ

– DM : : = < – dBm> | < – dBmV> | < – dBμV> | SC | MZ

units of voltage:

MV : : = <mV> | KZ | MS

UV : : = <μV> | Hz | US

units of frequency:

HZ : : = <Hz> | UV | US

KZ : : = <kHz> | MV | MS

MZ : : = <MHz> | – DM | SC

GZ : : = <GHz> | DM | DB

units of time:

SC : : = <sec> | MZ | – DM

MS : : = <msec> | KZ | MV

US : : = <μsec> | HZ | UV

<step> : : = UP | DN

Data Output From Analyzer

Output commands can be aborted during the output by addressing the analyzer to listen and issuing any legal command.

The final single character of any output will pull EOI true for data valid condition.

Syntax

Code	Syntax	Function
AT	<listen> AT[<entry> | <step>]	RF input attenuator
A1	<listen> A1	Clear write trace A
A2	<listen> A2	Max hold trace A
A3	<listen> A3	View trace A
A4	<listen> A4	Blank trace A
BL	<listen> BL	B – DL→B
B1	<listen> B1	Clear write trace B
B2	<listen> B2	Max hold trace B
B3	<listen> B3	View trace B
B4	<listen> B4	Blank trace B
CA	<listen> CA	Couples RF input attenuator
CF	<listen> CF[<entry> | <step>]	Center frequency
CR	<listen> CR	Couples resolution bandwidth
CS	<listen> CS	Couples center frequency step size
CT	<listen> CT	Couples sweep time
CV	<listen> CV	Couples video bandwidth
C1	<listen> C1	A – B off
C2	<listen> C2	A – B→A on
DA	<listen> DA [<entry>]	Sets display memory address
DB	see <units code>	
DD	<listen>[<address>] DD <binary value> <binary value> where <address> set by DA.	Writes two 8-bit binary bytes (O4 format) into display address selected.
DL	<listen> DL [<entry> | <step>]	Enables display line and places it at the value level entered.
DM	see <units code>	
DN	<listen> DN (also see <step>)	DATA step down
DR	<listen>[<format>][<address>] DR <talk>	Outputs the contents of designated display address onto the HP-IB data lines. The contents are formatted. Each DR increments display address by 1.
DT	<listen> DT <8 bit binary byte>	Establishes a character, in addition to <ETX>, to terminate a label (LB) entry or a title (KSE) entry. The character will not be stored in display memory when used in a label.
DW	<listen>[<address>] DW [<entry>]	Writes the value into the display address specified. Each value written increments address by one.
D1 D2 D3	<display size> : : = D1 | D2 | D3	Sets the display size for CRT graphics

Code	Syntax	Function
EE	<listen> EE	Allows operator to make an entry to the DATA buffer with the DATA number/units keyboard while EE is active.
EK	<listen> EK	Allows operator to change any active function value with the DATA knob while EK is active.
EM	<listen> EM	Replaces trace C addresses (3073-4095) with an end of memory word, 1044. Resets the display address to 3072.
EX	<listen> EX	Exchanges A and B traces.
E1	<listen> E1	Enables peak search.
E2	<listen> E2	Marker into center frequency.
E3	<listen> E3	Marker or Δ into step size.
E4	<listen> E4	Marker into reference level.
FA	<listen> FA [<entry> | <step>]	Start frequency
FB	<listen> FB [<entry> | <step>]	Stop frequency.
FS	<listen> FS	0-1.5 GHz span.
GR	<listen>[<display size>][<address>] GR <y value> where <y value> : : = <value> <delimiter> | <y value> <y value>	Plots successive values as amplitudes on CRT display, incrementing horizontal positions left to right by one for each value. Trace starts at x = 0 position.
GZ	see <units code>	
HD	<listen> HD	Holds or disables DATA entry and blanks active function CRT readout.
HZ	see <units code>	
IB	<listen> IB <exactly 2002, 8 bit binary bytes>	Inputs trace B beginning at address 1025. CR or LF must not occur after IB.
IP	<listen> IP	Instrument preset.
I1	<listen> I1	Selects SIGNAL INPUT 1, left input port.
I2	<listen> I2	Selects SIGNAL INPUT 2, right input port.
KS	<listen> KS	Shift front panel key functions.

Note: Only shift functions which allow entries are listed. All others have the general syntax:
<listen> KS <8 bit binary byte>

Code	Syntax	Function
KSE	<listen> KSE <ASCII or analyzer character string> <title terminator> where <title terminator> : : = <ETX> | <LF> | <character selected by DT command>	Title mode characters called from analyzer character set are displayed on top line of CRT display. Up to 64 characters can be input.
KSG	<listen> KSG [<entry>]	Video averaging on. Maximum sample size set by entry value.

SYNTAX SUMMARY

Code	Syntax	Function
KSJ	<listen> KSJ <value > < delimiter> each entry programs different DACS:	
	<value> HZ	L.S. VTO DAC
	<value> KZ	M.S. VTO DAC
	<value> MZ	YTO DAC
	<value> GZ	SCAN ATTEN/All DACS
	<step>	Steps all DACs by power of 2
KSP	<listen> KSP [<entry>]	Sets HP-IB address.
KSV	<listen> KSV [<entry>]	Frequency offset.
KSZ	<listen> KSZ [<entry>]	Amplitude offset.
KS =	<listen> KS = [<entry>]	Counter resolution.
KS<	<listen> KS< [<entry>]	SIGNAL INPUT 1 preamp gain.
KS>	<listen> KS< [<entry>]	SIGNAL INPUT 2 preamp gain.
KS,	<listen> KS'[<entry>] [<step>]	Input mixer level in 10 dB increments.
KS 123_{10}	<listen> <format> <address> KS 123_{10} <talk> (for HP 9825A, KS 123_{10} is KSπ)	Outputs up to 1001 words of display memory beginning at the address specified. Words are in format specified (use format O2 if KS 125_{10} or IB are to be used). Words separated by LF, last LF sets EOI true. Output may be terminated at any time with a <go to local>.
KS 125_{10}	<listen> <address> KS 125_{10} <up to 2002 eight bit binary bytes> (for HP 9825A, KS 125_{10} is KS→)	Inputs up to 1001 display memory words (two bytes per word), beginning at address specified.
KZ	see <units code>	
LB	<listen>LB<character string> <label terminator> where <label terminator> : : = <ETX> | <character selected by DT;command >	Writes specified characters on the CRT display. First character appears at current CRT beam position. See PA and PR.
LG	<listen> LG [<entry>]	Enters LOG SCALE.
LL	<listen> LL	Provides a lower left x-y recorder output voltage at the rear panel for the duration while LL is active. See UR.
LN	<listen> LN	Linear SCALE.
LØ	<listen> LØ	Display line off.
MA	<listen>[<format>] MA <talk>	Outputs the marker amplitude onto the HP-IB DATA lines according to the format.
MCØ	<listen> MCØ	Marker counter measurement off.
MC1	<listen> MC1	Marker counter measurement on.
MF	<listen> [<format>] MF <talk>	Outputs the marker frequency onto the HP-IB DATA lines according to the format.
MS	see <units code>	

Code	Syntax	Function
MT0	<listen> MT0	Signal track off.
MT1	<listen> MT1	Signal track on.
MV	see <units code>	
MZ	see <units code>	
M1	<listen> M1	Marker off.
M2	<listen> M2 [<entry> | <step>]	Enables single marker, MARKER normal mode. Marker moves to frequency position of entry value in Hz. Entry must be positive.
M3	<listen> M3 [<entry> | <step>]	Enables second marker. Second marker moves to differential frequency position of entry value in Hz. Entry may be positive or negative.
M4	<listen> M4 [<entry> | <step>]	Enables marker zoom. Marker moves to frequency position of entry value in Hz. Step up or down changes span.
OA	<listen>[<format>] OA <talk>	Outputs the active function value.
OL	saving instrument state: <listen> OL <talk> <80,8 bit binary bytes>	Outputs coded instrument state information into the 80 binary variables.
	recalling instrument states: <listen> <same 80 bytes>	Recalls the instrument state. The first byte transferred to the analyzer establishes the recall mode.
OT	<listen> OT <talk> <32 strings>	Outputs all CRT annotation as strings. Strings are from 0 (null) to 64 characters long. Each string terminated with a <CR> <LF>. Last string terminated with an EOI upon <LF>.
		Output Formats:
O1	<listen> O1	ASCII number in display units.
O2	<listen> O2	Two 8 bit binary bytes.
O3	<listen> O3	ASCII number in parameter or instrument units.
O4	<listen> O4	One 8 bit binary byte. The two bytes form a single entry as follows: first byte: xxxxbbbb — second byte: bbbbbbbb 12 bit word MSB
PA	<listen>[<display size>] [<address>] PA <xy pair> where <xy pair> : : = [PU | PD] <value> <delimiter> <value> <delimiter> <xy pair> <xy pair>	Plot absolute draws vectors to x and y entries. PU and PD determine whether vector is displayed or blanked. Entries must be in positive display units.
PD	<listen> PD	Pen down turns beam on.

Code	Syntax	Function
PR	<listen>[<display size>][<address>] PR <xy pair> (see PA)	Plot relative draws vector relative from the last absolute position. Entries can be positive or negative.
PS	<listen> [<address>] PS	Display program skips to next page of memory from address specified.
PU	<listen> PU	Pen up turns beam off.
RB	<listen> RB [<entry> | <step>]	Resolution bandwidth.
RC	<listen> RC 0|1|2|3|4|5|6|7|8|9	Recalls instrument states 0 to 9.
RL	<listen> RL [<entry> | <step>]	Reference level.
R1	<listen> R1	Resets SRQ to allow only SRQ 140.
R2	<listen> R2	Allows SRQ 140 and 104
R3	<listen> R3	Allows SRQ 140 and 110
R4	<listen> R4	Allows SRQ 140 and 102 R2, R3 and R4 not mutually exclusive
SC	see <units code>	
SP	<listen> SP [<entry> | <step>]	Frequency span.
SS	<listen> SS [<entry> | <step>]	Center frequency step size.
ST	<listen> ST [<entry> | <step>]	Sweep time.
SV	<listen> SV 1|2|3|4|5|6	Saves instrument states in register 1 through 6.
SW	<listen>[<address>] SW	Display skips to next control word from address specified.
S1	<listen> S1	Sweep continuous.
S2	<listen> S2	Sweep single.
TA	<listen> [<format>] TA <talk> <1001 Words>	Outputs 1001 trace amplitude values for trace A, beginning with the leftmost trace point.
TB	<listen> [<format>] TB <talk> <1001 Words>	Outputs 1001 trace amplitude values for trace B, beginning with the leftmost trace point.
TH	<listen> TH [<entry> | <step>]	Threshold level.
TS	<listen> TS	Take a sweep. Analyzer does not handshake until sweep is complete, and marker, if on, is placed on trace.
T0	<listen> T0	Threshold level off.
T1 T2 T3 T4	<listen> T1 | T2 | T3 | T4	Selects trigger mode T1 free run T2 line T3 external T4 video
UP	<listen> UP	DATA step up.
UR	<listen> UR to turn off UR and LL <listen> UR LL LL	Provides an upper right x-y recorder output voltage at the rear panel while UR is active.
US	see <units code>	μsec
UV	see <units code>	μvolt
VB	<listen> VB [<entry> | <step>]	Video bandwidth.

Index